Three Essays on the Management and Economics of Blockchain-Based Systems

Dissertation
submitted to the Faculty of Business,
Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor der Wirtschaftswissenschaften, Dr. oec.
(corresponds to Doctor of Philosophy, PhD)

presented by

José Parra Moyano
from Spain

approved in September 2019, at the request of
Prof. Dr. Karl Schmedders
Prof. Dr. Claudio J. Tessone

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, 18.09.2019

Chairman of the Doctoral Board: Prof. Dr. Steven Ongena

Contents

Abstract

This thesis compiles three papers on different topics in blockchain-based information systems. The first paper presents a blockchain-based information system to reduce the costs and improve the customer experience in the Know-Your-Customer (KYC) process that financial institutions are obliged to conduct. The system shows how to use blockchain technology to reduce and share out the cost of the KYC process among a consortium of financial institutions without compromising the privacy of customers and institutions. The second paper improves the system presented in the first paper by combining it with a distributed database, which allows for a more flexible and decentralized architecture. Further, the improvements in the second paper allow for dynamic updates in the status of the customers. The third paper revises the existing assumption that states that the probability of miners in the bitcoin system finding a valid block is constant within each block and governed by the Poisson distribution. By means of a multinomial logit model, the third paper shows that the probability of a miner finding a valid block increases with the time elapsed since the miner started mining the block. Further, the third paper postulates that a possible explanation for this phenomenon is that the probability of miners in the bitcoin system finding a valid block is governed by the negative hypergeometric distribution.

To Coral:

May you live a happy life.

Acknowledgments

I would like to sincerely thank the following persons, who have directly or indirectly made this dissertation possible.

Prof. Dr. Karl Schmedders, my supervisor, for his immense support, valuable guidance, breath-taking discussions, and infinite generosity; *I hope that I can live up to what I have learned from you.*

Prof. Dr. Claudio J. Tessone for his supervision, advice, and time; *Only honest candor will make us advance.*

The coauthors of my research—Dr. Gregor Reich, Prof. Dr. Omri Ross, and Tryggvi Thorodsen—as well as Dave Brooks, Marcel Bühler, Dr. Carlos Riquelme Ruiz, and all the members of the Chair of Quantitative Business Administration of the University of Zurich for their ideas, time, and consideration; *You honor the words "team-work" and "science".*

José, my father, a giant upon whose shoulders I stand. Carmen, my mother, a role model in all the matters of the heart. Carmen and Pedro, my sister and my brother, my oldest friends and allies. *To all of you: I am who I am because you are who you are.*

Raquel, my dear wife, for her constant love, precious patience, and tremendous understanding, especially during the last phase of this dissertation; *You enlighten the darkness of the night, give me shelter in the storm, and strengthen me when everything seems lost. I will always be there for you.*

Part I

Introduction

Introduction

The seminal work by Nakamoto (2008) introduced the concept of "bitcoin", an electronic currency and economic system based on a technology that has been given the name of "blockchain". The main innovation introduced by this system was that it offered a solution to the "double-spending" problem that was intrinsic to previous decentralized, digital currencies. This problem emerged from the fact that without a third party or central authority overlooking the system, users of electronic currencies could spend the same units of the currency twice, engaging in fraudulent behavior. Nakamoto (2008) designed the bitcoin system to be sustained by nodes that communicate with each other in a peer-to-peer network. Some of these nodes are called "miners" in the blockchain jargon. Miners conduct a trial-and-error process that can result in either success or failure. Success in this process results in a new block (a piece of information about the new transactions that are accepted in the system) being appended to the blockchain (a ledger containing all the past information regarding the transactions made in the system). This new block contains a transaction that awards the winning miner a reward in the form of bitcoins. These bitcoins compensate the mining costs of the miner, which are the fixed cost of the mining hardware and the variable electricity costs associated with the mining activity. In order to conduct this trial-and-error process, miners devote hash power to the network. The higher the hash power that a miner devotes to the system, the higher her probability of success in the mining process. The way Nakamoto (2008) suggested solving the double-spending problem was by making the cost of conducting a double-spending attack so high that the expected return of a double-spending attack would result in a negative monetary amount. Since a dishonest miner wishing to conduct a double-spending attack on the network needs to carry out the trial-and-error mining process faster than all the honest miners together, the dishonest miner needs to devote more hash power to the network than all the honest miners, which implies a very high monetary cost for the dishonest miner. The cost of such an attack is so high that it is expected to not be compensated by the amount in bitcoins that can be spent twice in a double-spending attack.

By solving the double-spending problem, Nakamato (2008) was able to propose an economic system that was anonymous, fully peer-to-peer, and that required no trusted third party in order to properly

function. Further, this system was able to create and distribute value without the need of centralized third parties. Closely following the publication of Nakamoto's work (2008), bitcoin drew the attention and admiration of libertarians and so-called crypto-anarchists, since—as described by Karlstrøm (2014)—it allowed them to escape the centralized, highly regulated traditional economic system. For many hackers and criminals, bitcoin was also attractive, since it represented a way to conduct illicit activities in an anonymous manner. In fact, Foley et.al. (2019) show that in 2009 the percentage dollar volume of illegal bitcoin user transactions reached 85 percent of the whole dollar volume of bitcoin user transactions. Bitcoin seemed to fulfill the prediction made by Milton Friedman in 1999 with regard to the type of money that was required if the Internet was to unleash its real potential: "I think that the Internet is going to be one of the major forces for reducing the role of government. The one thing that's missing, but that will soon be developed, is a reliable e-cash, a method whereby on the Internet you can transfer funds from A to B without A knowing B or B knowing A. [...] That kind of thing will develop on the Internet and that will make it even easier for people using the Internet. Of course, it has its negative side. It means the gangsters, the people who are engaged in illegal transactions, will also have an easier way to carry on their business." (Friedman, 1999). During the initial years following its introduction, financial institutions, governments, and respected economists would not only not back bitcoin, they would come out clearly against it. Examples of such opposition include Krugman (2013), whose work bears the title "BitCoin is evil", Stross (2013), whose work is entitled "Why I want Bitcoin to die in a fire", and Basu (2014), of the World Bank stating that "Bitcoin is a naturally occurring Ponzi scheme". Kurgman (2013), Stross (2013), and Basu (2014) criticize many aspects of the bitcoin system, including the electricity costs required to keep it secure, its anonymity, its volatility, the lack of regulation, the impossibility of implementing monetary policy on the system, and the deflationary aspects of the currency. In this vein, and representing well the negative opinion of bitcoin, Nouriel Rubini stated that "Bitcoin isn't a currency. It is a Ponzi game and a conduit for criminal/illegal activities." (Roubini, 2014).

However, and despite these statements and analyses, both scholars and practitioners realized that blockchain technology had certain aspects that could be interesting to incorporate into other systems. Such aspects included the ability to create and transmit value between peers in a decentralized manner, the ability to achieve consensus without the need of a third party, the ability to have immutable, distributed ledgers to store information, and the ability to design new forms of corporate organization—that is to say, decentralized autonomous organizations, whose governance rules are specified in the blockchain (Beck, 2018). Further, they realized that blockchains can also contain "smart contracts", which are programs stored on the blockchain that run as implemented without any risk of downtime, censorship, or fraud (Buterin, 2014). Additionally, as stated by Lindman et al. (2017), blockchain technology was recognized as having the potential to become a valuable enabler of economic and social transactions, for instance as a general-

purpose digital asset ownership record. Beck et al. (2017) describe how the financial sector has been leading the way in developing blockchain applications and business models, but also how companies in industries such as shipping, transportation, healthcare, and entertainment are actively using blockchain applications to coordinate the movement of products, facilitate the creation of e-health records, and securely manage original entertainment content. Summarizing, after the backlash from economists and institutions against bitcoin in the years following its introduction, many properties of the blockchain are being used in 2019 to improve processes in existing industries, allow competitors to collaborate in certain aspects of their business processes, and increase economic growth by enabling new business models.

In this context, Chapter 1 presents a blockchain-based information system that uses blockchain technology (called "distributed ledger technology" (DLT) in the article due to the lack of consensus regarding the nomenclature back when the chapter was written) to reduce the cost of the Know-Your-Customer (KYC) due diligence process conducted by financial institutions. The KYC process is a highly regulated verification process that needs to be carried out by financial institutions before starting to work with any client. This process is very similar across all the financial institutions operating under the same jurisdiction, occurs in a parallel or consecutive manner, and generates costs of up to USD 500 million per year per bank (Thompson Reuters, 2016). In the system presented in Chapter 1, the KYC verification process is only conducted once for each customer, regardless of the number of financial institutions with which that customer intends to work. In this system, the result of the KYC process (a document accepting or rejecting the client as a viable client according to the jurisdiction) can be securely and privately shared by customers with all the financial institutions that they intend to work with, avoiding the need for each single financial institution to repeat the KYC process. This system allows for efficiency gains, cost reduction, improved customer experience, and increased transparency throughout the process of onboarding a customer. Chapter 1 contributes to the literature by presenting a use case of blockchain technology to address an existing problem in the financial industry. Further, it shows how the role of economic incentives is crucial for the design and correct implementation of blockchain-based systems and how the design of the incentives structure plays a vital role in the properties that emerge from blockchain-based systems. While the piece makes a solid contribution to the literature and was published under Parra-Moyano and Ross (2017), it is a static system that allows for no updates in the status of a customer, requires a central authority (the national regulator) to conduct compensations between financial institutions, and needs the customers of financial institutions to be responsible for the handling and maintenance of their own documents, which increases their workload compared to the KYC case without a blockchain.

Chapter 2 addresses the issues left open in Chapter 1. Specifically, it presents a system that makes it possible for financial institutions to dynamically update information related to their customers and

disseminate this information among all the financial institutions that participate in the system. Further, the system presented in Chapter 2 incorporates the distributed database architecture of Siegenthaler and Birman (2009a and 2009b) into the design of the KYC system. The incorporation of a distributed database architecture to securely and privately store the data of the customers outside of a blockchain significantly increases the flexibility of the system presented in Chapter 1 (it implies a much simpler architecture) while keeping the privacy standard intact. Further, it eliminates the need for the central authority to manage the payments, something that was required in the system presented in Chapter 1. Chapter 2 also presents the programmed artifact described in Chapter 1 and Chapter 2, and makes it available for scholars and practitioners to use for their own ends. The major contribution of Chapter 2 lies in its combination of the distributed database architecture of Siegenthaler and Birman (2009a and 2009b) and the blockchain technology introduced by Nakamaoto (2008). Using a distributed database architecture to privately store data and using the public blockchain to manage the reading permissions on the database significantly reduces the complexity and costs of the system while maintaining its benefits intact. Chapter 2 will be published as Parra-Moyano, Thoroddsen, and Ross (2019).

Chapter 3 studies the bitcoin protocol, which is the inspiration behind all the proof-of-work blockchain protocols that sustain blockchain-based systems. Thus far, it has been assumed in the literature that the success of the mining trial-and-error process follows the Poisson distribution. From this assumption it emerges that the success probability of a miner remains constant throughout the mining process for each block. Chapter 3 describes a series of observations that contradict this assumption, stating that the probability of a miner finding a valid block increases with the time that elapses since the moment at which that miner starts mining a particular block. Chapter 3 postulates that a possible explanation for this phenomenon is that the probability of a miner finding a valid block in the bitcoin network (the rate of successful trial-and-error mining processes) is governed by the negative hypergeometric distribution. In order to test if the probability of winning increases with time, Chapter 3 models the mining process as a race, in which miners compete to find the next valid block, and by means of a multinomial logit model with the same structure as the one used by Bolton and Chapman (1986) shows that we should reject the idea that time does not increase the winning probability of miners for a particular block in a manner that is proportional to a miner's size. The dataset used to compute the model requires many assumptions since the econometrician cannot observe many of the variables required for the computation of the model and therefore needs to infer them. For this reason, and given the fact that the result of this chapter might have serious implications for proof-of-work systems, Chapter 3 concludes with a call to the mining community to publish the data that they have and that the econometrician cannot observe, in order to facilitate a definite answer to the clarification of this phenomenon.

Summarizing, this thesis contributes to the literature by studying, designing, and developing a blockchain-based system that addresses an open issue in the banking industry. By doing so, this thesis presents how to use economic incentives to design a blockchain system such that all the agents in the system behave as they are meant to behave. Additionally, this thesis makes an in-depth study of bitcoin mining, showing that despite the useful attributes of blockchains, some of its fundamental aspects might have not yet been fully understood—neither by scholars nor by practitioners—and that further reflection and analysis is required before blockchain technology is mature enough to unleash the economic growth that it clearly promises.

Part II

Three Essays on the Management and Economics of Blockchain-Based Systems

Essay 1

KYC Optimization Using Distributed Ledger Technology

KYC Optimization Using Distributed Ledger Technology

José Parra-Moyano
University of Zurich
Switzerland
jose.parramoyano@uzh.ch

Omri Ross
University of Copenhagen
Denmark
omri@di.ku.dk

November 2017

Abstract

The know-your-customer (KYC) due diligence process is outdated and generates costs of up to USD 500 million per year per bank. The authors propose a new system, based on distributed ledger technology (DLT), that reduces the costs of the core KYC verification process for financial institutions and improves the customer experience. In the proposed system, the core KYC verification process is only conducted once for each customer, regardless of the number of financial institutions with which that customer intends to work. Thanks to DLT, the result of the core KYC verification can be securely shared by customers with all the financial institutions that they intend to work with. This system allows for efficiency gains, cost reduction, improved customer experience, and increased transparency throughout the process of onboarding a customer.

Keywords: Blockchain, Information Systems, KYC, DLT

Note: A version of this paper has been published as Parra Moyano, J. and Ross, O., KYC Optimisation Using Distributed Ledger Technology, Business & Information Systems Engineering, 59 (6), 411-423, 2017.

1.1 Introduction

The increased regulatory cost incurred due to the know-your-customer (KYC) verification process in banking is one of the largest challenges that the banking sector is currently experiencing. The yearly direct costs that financial institutions need to cover in order to meet their obligations in terms of KYC are estimated, in a recent survey by Thompson Reuters (2016), to average USD 60 million. This cost can be further augmented by the fines levied on financial institutions due to their misconduct with regard to anti-money-laundering (AML) and KYC regulations. According to the head of Strategy and Risk at the Hong Kong Securities and Futures Commission, "KYC and AML stand out [for a bank to] as a pretty significant inefficiency and problem case [...] tallying up the fines [for a bank to] 10 billion or more US dollars" (Benedict N. Nolens, at the MIT Technology Review Emtech conference, 2016). And the sources of additional costs do not stop here, as financial institutions are not allowed to conduct any business with corporate entities that have not yet completed the full KYC process. Since that process is long, and tends to lengthen with the size of the corporate entity concerned, the starting point of a given business relationship between a customer and a financial institution is usually delayed, which represents opportunity costs for both parties. Indeed, corporations need to verify all their subsidiaries before being granted KYC verification, and this is a laborious task for them. Therefore, it comes as little surprise that the abovementioned survey indicates that 89% of customers do not have a good KYC experience.

The aim of this paper is to propose a new approach to the KYC verification process. We introduce a system, based on DLT, that proposes a solution to the increased costs of the KYC process and the lack of customer satisfaction. The key reason for using DLT is that it allows us to observe the KYC cost structure at an aggregate level for all the financial institutions operating in a jurisdiction and to tackle the inefficiencies that emerge from the duplicated conduct of similar tasks by all participating institutions (i.e., DLT allows us to render the execution of duplicated tasks completely unnecessary, and this delivers far greater cost savings than would any effort to merely make these duplicated tasks more cost efficient). Specifically, DLT enables the creation of a chronological, decentralized, interbank ledger in which financial institutions that need to conduct the same KYC verification tasks for that customer can verify the result of the process that has already been conducted for that customer, thus avoiding conducting duplicated KYC verification tasks. Moreover, the use of DLT allows the cost of the KYC process to be shared proportionally among the financial institutions that work with a specific customer. In particular, the system allows customers to carry out the full KYC process with only one financial institution, and later on to share the result of that KYC process with any other financial institution that they intend to work with. The DLT acts as a "single point of truth", understood as the only source of

information, accepted by any involved party should conflict occur.

The main improvement of the proposed system over the current system is that the KYC process only needs to be carried out once by each customer, rather than once by each institution working with that customer. This reduces the aggregated cost of the KYC process as a whole in a jurisdiction without compromising the security of the system, respects the privacy of the participants, and increases transparency in case of a conflict. Additionally, the use of the public key of a customer as a reference point for an immutable exchange of information across participating institutions serves as a basis for interbank collaboration. The use of DLT reduces the aggregate cost of KYC and this is the main conceptual contribution of this paper. In Section 1.2 we explain the KYC process, and relate it to work that has already been carried out with regard to optimizing KYC costs. Section 1.3 offers an overview of DLT and examines its potential for resolving the current problems of the KYC process. In Section 1.4 we show how we have applied design science research to solve the problem at hand. In Section 1.5 we describe and analyze the prototype solution and the economic mechanisms that need to be put in place in order to ensure a well functioning system. In Section 1.6 we discuss three possible implementations of this solution. Section 1.7 concludes.

1.2 The Current KYC Process

The KYC process is part of the growing regulation of the financial industry that began with the Money Laundering Control Act of 1986 (see USA 1986) and has been growing extensively since in the form of further, ongoing regulation aimed at precluding either money laundering or the funding of terrorist activity (see USA 1988, 1992, 1994, 1998, 2001, 2004). Financial institutions are obliged by regulators to onboard their customers before conducting any activity with them, in order to avoid working with customers that pursue either of the aforementioned illicit activities. The KYC process consists of an exchange of documents between the customer and the financial institution that intend to work together. The process includes the collection of basic identity information from all beneficiaries to check for illicit activity and "politically exposed persons." The process also includes risk management with regard to onboarding new customers, the monitoring of transactions, and specific customer policies for banks. The process is costly for financial institutions and may expose them to large fines if it is not conducted in accordance with the existing regulations (e.g., HSBC was fined USD 1.92 billion when it was discovered that Mexico's Sinaloa cartel and Colombia's Norte del Valle cartel had laundered USD 881 million through the bank (Viswanatha and Wolf 2012), and ING Bank paid USD 619 million in fines for violating sanctions against a variety of countries (Freifeld 2012)).

The KYC process is initiated when a customer intends to work with a financial institution. Chronologically, the customer and the financial institution agree on the terms of a relationship. Then, the customer sends the required documents to the financial institution in order to enable the institution to conduct the KYC verification process. The financial institution analyzes the documents and generates an additional, internal document that serves as the certification that assures regulators that this customer has been either validated or rejected and that the KYC process has been properly conducted. This process is repeated every time the customer intends to work with a new financial institution. In the current setting, every time a customer initiates a relationship with a financial institution the costs of the KYC verification process recur. Figure 1 shows an example case that illustrates the process that occurs when a customer intends to work with three different financial institutions. This example case shows how, for this single customer, the exchange of documents and the core KYC validation must be undertaken three times, such that the total costs that are generated by this customer are three times those of a single KYC process. At this point, it is important to differentiate between the "core KYC verification process", which is the minimum KYC verification that all financial institutions are obliged by law to conduct, and additional, bank-specific processes. While further documentation can be asked for by each financial institution to create an "additional aura of information" for every customer, our solution focuses solely on the core KYC verification process, which is that shared by all the financial institutions in a jurisdiction.

The growth of regulation and changes to technology, as well as the financial crisis of 2007, have created opportunities for companies, working in a field referred to as "regtech", that aim to use technology to improve the implementation of regulations. The term "regtech" comes from the combination of the words "regulation" and "technology". These opportunities are especially significant within the domain of KYC (see Memminger et al. 2016; Arner et al. 2016). Arasa and Ottichilo (2015) conduct an analysis of the cost of KYC based on the complexity level of the compliance required for the case of commercial banks in Kenya, establishing four variables that explain 78.3% of the compliance requirements. Soni and Duggal (2014) look into using big data analytics to reduce risk for institutions conducting the KYC process. Colladon and Remondi (2017) work on different approaches to using cluster analysis over a network of customers and potential customers to identify suspicious financial operations and potentially criminal activities. They do so by mapping relational data and using predictive models over an internal transactions database involving data from over 33,000 financial operations. A survey of the latest regulatory requirements and a history of KYC and AML processes can be found in Ruce (2011). KYC can be improved by, for example, improving auditors' effectiveness in assessing KYC and AML practices. A case study in the context of Luxemburg is provided by Smet and Mention (2011) and reveals that audit effectiveness could be increased and information asymmetries reduced by an ISO standard for an internal control assessment model for KYC. The current paper aims to deliver an additional improvement by using

DLT to reduce the aggregate cost of the KYC process and distribute these lower costs proportionally among the financial institutions participating in the system. Tackling the cost of the KYC process from the aggregated perspective (i.e., as the sum of the individual costs of each financial institution) and using DLT to reduce this aggregate cost is the main contribution of this paper.

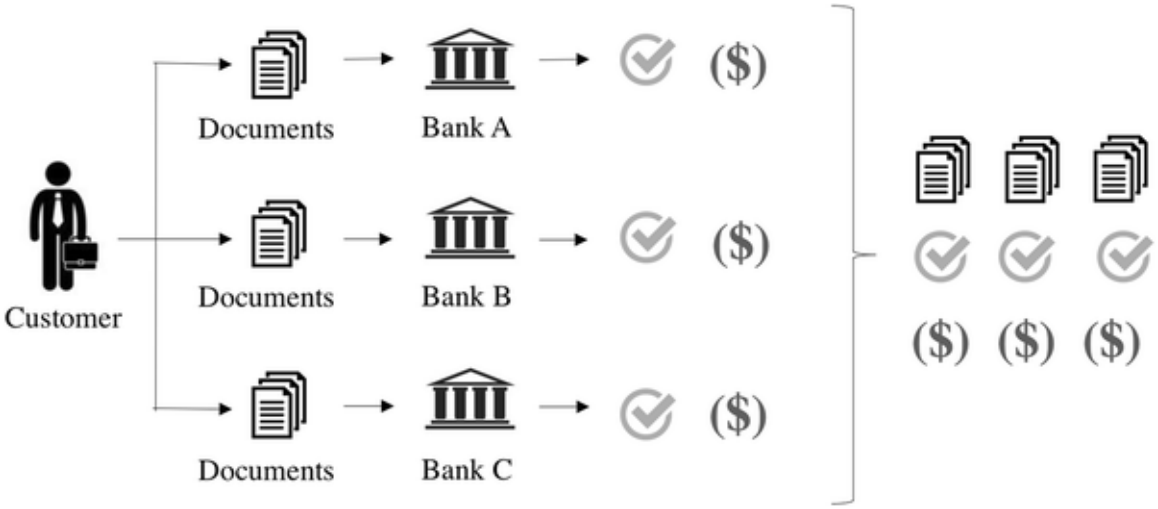

Figure 1: Current process and cost structure of KYC

1.3 Blockchain Technology

DLT, such as blockchain technology, has gained prominence thanks to the widespread use of the cryptocurrency Bitcoin. Bitcoin, introduced by Nakamoto (2008), was the first working cryptocurrency that was not owned by a central authority. While DLT was originally used to provide a new way of creating money and transferring it via the Internet, the technology can also be used to run and govern decentralized systems by means of smart contracts. Smart contracts are computer protocols that facilitate, verify, or enforce predefined clauses whenever a set of conditions is given. As described by Szabo (1997), the intention of using smart contracts is to embed them in a whole range of properties that are valuable and controlled by digital means. Since Nakamoto's seminal work (Nakamoto 2008), new instances that propose the use of DLT for a range of novel purposes have emerged. One of these is "Ethereum", which is a platform upon which whole decentralized applications may be run (see Wood 2016). Many papers, including Peters and Panayi (2015) and Harvey (2016) discuss the blockchain from a technical perspective.

While transactions in the Bitcoin blockchain can include small scripts that define output spending conditions, such as the requirement that a transaction be signed by two keys instead of one, the Ethereum blockchain can be seen as a Turing complete virtual machine that can run code in several programming languages and therefore run the smart contracts stored in it (see Glaser 2017).

Glaser (2017) provides a solid ontological development of blockchain systems concepts and defines a common set of blockchain components and relationships. This analysis serves as a framework and basis for assessing the implications of blockchain solutions in an academic or economic context. Further, it introduces the perspective of a pervasive decentralization of multiple layers of digital infrastructure by blockchain technology. Specifically, Glaser (2017) defines and describes two layers of code – namely, the fabric layer and the application layer. The term fabric layer denotes the system's code base, which embraces communication, the public-key infrastructure, the software that constructs and maintains the database, and the execution environment of the system. Whoever develops and maintains the fabric layer controls the functioning of the system. Ultimately, the fabric layer defines the governance type of the system, which can be the only dimension of the fabric layer, and that can be public, permissioned, or hybrid. Nevertheless, and as described by Glaser (2017), one important characteristic of blockchain systems is that they do not allow for a differentiation between users and user management modules, which implies that all the users have complete transparency when reading the transactions and the smart contract code deployed.

The application layer comprises the application logic of the services implemented in the form of smart contracts. The application layer encompasses three dimensions – namely, the ecosystem closedness, the value linking, and the market type. The closedness of the ecosystem refers to the extent to which the system needs to interact with other structures that are outside of the blockchain-based framework – that is, with other trusted interfaces. Since the decentralization of control ends at the boundaries of the blockchain-based system, the more closed the system is, the higher the leverage of a blockchain-based solution. The value linking of the system refers to the intrinsic value of the tokens that are exchanged between parties within the system. Glaser (2017) suggests four possible ways in which value is assigned to the tokens of a system – namely, being the token a community currency, being seen as debt or equity by the participants of the system, being backed by a commercial bank, or being backed by a central bank. The last dimension of the application layer is market type, which describes the nature of the market in which the blockchain-based solution is framed.

The European Security and Markets Authority (2016) sets out the possible benefits of DLT applied to securities markets, discusses the possible shortcomings of and challenges to those benefits, and analyzes the relevant regulatory framework, with a focus on the main EU legislation relevant to potential applications of DLT in securities markets. While the Authority focuses on the securities market, it provides a DLT-solutions governance framework that can be very similar to the governance framework required by the solution proposed in this paper. Specifically, it suggests that for the interbank context of

securities markets, a permission-based system can be of great value. Further, the Authority claims that such a system would allow for governance of the interaction between the system's participants, paying special attention to the liabilities of each participant, correction mechanisms, and even penalties in the case of infringement of the rules.

The European Central Bank (2012) defines and classifies virtual currency schemes based on their observed characteristics. Depending on the interaction of the virtual currency schemes with traditional money and the real economy, the Bank classifies them into three types: Type 1, which refers to closed virtual currency schemes, which operate in the same way as do virtual currencies used in online gaming; Type 2, virtual currency schemes with a unidirectional flow (usually an inflow), meaning that there exists a conversion rate for purchasing the virtual currency; and Type 3, virtual currency schemes that have bidirectional flows. The World Economic Forum (2016) analyzes the current phase of the disruptive innovation work that is being conducted in terms of DLT in the financial sector, first looking at how blockchain can reshape financial services, and then studying the role of financial institutions in building digital identity. The Forum (2016) concludes that DLT can enable the design of new systems or improve existing ones, by automating processes, reducing settlement time, reducing costs, reducing operational risk, providing central authority disintermediation, and offering real-time settlement. Egelund-Müller et al. (2017) look into the construction of an automated financial system, with multiple counterparties, that can run a variety of complex financial derivatives, including settlement, directly on DLT.

1.4 Design Science for KYC Optimization

According to Hevner et al. (2004), the objective of design science research (DSR) is to produce a technology based solution – in the form of a viable artifact – that solves a relevant business problem. In the context of a hackathon organized at the IT University of Copenhagen, we collaborated with the Nordic financial services group Nordea Bank AB to study the inefficiencies and costs related to the KYC process, and analyzed if this process could be improved by means of a DLT-based solution. During these four days we were confronted with the aforementioned reality of KYC inefficiencies, and transformed the existing problematic into the following research question:

"Can a DLT-based solution reduce the cost of the KYC process for financial institutions and improve the customer's experience?"

In order to answer the research question and to design an effective artifact that solves the problem at hand within the corporate and regulatory context, we followed Hevner et al. (2004)'s DSR approach

and focused on its three components (environment, IS research, and knowledge base). To strengthen the utility, quality, and efficacy of the proposed solution, we also considered the DSR process based on Peffers et al. (2007)'s approach, which synthesizes design science processes from Information Systems (IS) and other disciplines. This process is subdivided into five sub-steps: problem identification, objective definition, design and refinement of the artifact, demonstration of the artifact, and evaluation of the artifact. The last three steps of the process need to be repeated recursively in a loop in order to gather feedback from the environment and to refine the artifact according to that feedback. Both the approach and the process are summarized in Figure 2.

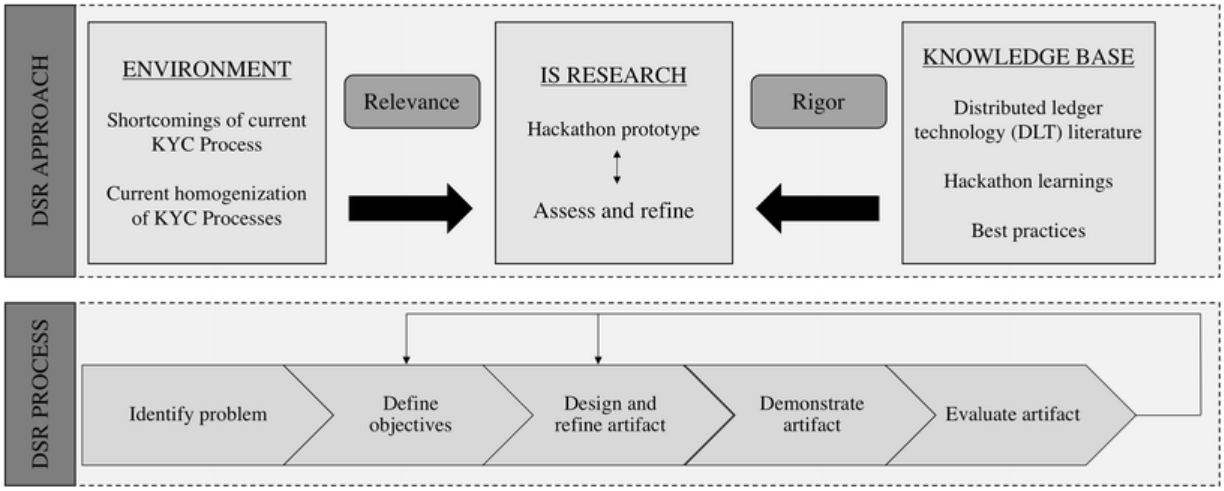

Figure 2: DSR approach and DSR process. Source: Authors' own illustration adapted from Hevner et al. (2004) and Peffers et al. (2007)

Nordea Bank AB, representing the corporate environment, expressed the need for improvement in the KYC process. They provided us with information concerning the applied difficulties of conducting the process and pointed out its main pain sources. This enabled us to identify the problem and define our objective (previously formulated in the form of our research question): use a DLT-based solution to reduce the cost of the KYC process for financial institutions and improve the customer's experience. In order to better understand the environment, we researched the existing KYC literature, paying special attention to efforts made in recent years to homogenize the KYC process and increase its efficiency without compromising security. Further, we held various exchanges with experts in the field (lawyers, practitioners, and experts) regarding best practices in KYC. During these exchanges, it became clear that the system proposed would need to fulfill three conditions if it was to be accepted by the participants. First, it would need to enable its users to obtain a tamper-proof record of the KYC process in the case of conflict. Second, it would have to reduce the costs of the current KYC process and distribute the remaining costs in a proportionate manner among the participants of the system. Third, the system would need to not

compromise the responsibility of banks with regard to conducting the KYC process. The combination of the environment's needs and our knowledge base constituted the grounds for our IS research, which yielded the first version of our artifact, a version that we continued to refine over several months based on ongoing dialog with, and feedback on the artifact from, KYC practitioners. With the problem identified and our objective defined (see above), the first design and refinement phase of the artifact was conducted, taking into account the feedback and validation of KYC practitioners, as well as the insights with regard to DLT from our knowledge base and the KYC experience from the environment. The first demonstration of the artifact took place during the Nordic Blockchain Summit, at which it was awarded first prize, receiving the majority of the votes of an audience of over 300 practitioners from the senior corporate management level. The first evaluation phase involved various informal working sessions with KYC practitioners who studied the artifact in terms of its relevance and viability, which helped us to learn more about the specific requirements of the participants.

After the first design, refinement, demonstration, and evaluation phases, we undertook a second loop of refinement, demonstration, and evaluation, following the DSR process described in Figure 2. The second loop incorporated the feedback of five senior executives from the banking sector, a lawyer, and two senior government officials, with whom we conducted several working sessions to explore various implementation possibilities of the solution here proposed. Their feedback was related to the need for interbank collaboration and for cooperation with the national regulator, as well as the need to launch the process in a single, relatively small country (that can amend the required regulations efficiently and quickly), to ensure that the system functions correctly. This feedback round made us aware of the need to initially propose the solution at a national level, moving on to a solution that would encompass a range of countries only later. From these working sessions, we also learned about the central role of the national regulator as the cornerstone of such a DLT-based solution, about the need to identify the individuals involved at each step of the KYC approval process, and about the importance of keeping all the documents of a specific customer on a secure local storage facility with only the hashes of each document stored on the DLT (in order to facilitate the tracing of past activity while ensuring that banks still know their customers and can effectively protect customer privacy with regard to cyber attacks). These points were influential in our decision to assign to the national regulator the role of maintaining the system.

1.5 The Redefined KYC Process

The IS suggested in this paper to solve the current inefficiencies of the KYC process relies on the following three assumptions: First, a group of financial institutions, working in the same country and therefore

obliged to respect the same KYC regulations, agrees on the standards for granting core KYC verification to a customer. Second, all the financial institutions that collaborate in the system agree on the average costs of conducting a core KYC verification process. This cost might of course depend on the complexity of each individual customer, based on predetermined parameters (e.g., client size, volume of documents exchanged, etc.). Third, the national regulator maintains the system and approves financial institutions to work with the system in order to conduct a more efficient and transparent KYC verification process. These three assumptions are necessary to ensure a correct incentive structure across the participating financial institutions.

Further, we define a set of four conditions that must be fulfilled by the artifact. It must ensure the proportional sharing of the cost of conducting the core KYC verification process; maintain the privacy standards of the KYC process as they are today; ensure that no institution can claim compensation without conducting that core process; and ensure that no institution can become a free rider and avoid paying for using the information generated by other member institutions. The proportionality condition ensures that the costs are shared proportionally. The irrelevance condition ensures that the financial institution that conducts the core KYC verification process does not have an incentive to prefer that another institution conducts the core KYC verification process and vice versa. The privacy condition ensures that the financial institutions that work in the system cannot know with which other financial institutions the customer is working, unless the customer reveals that information (privacy is required among financial institutions). The no-minting condition ensures that no financial institution can simulate having conducted a core KYC verification process in order to be compensated by other institutions for work that it has not done. These conditions are summarized in Table 1.

Name	**Description**
Proportionality	Ensure that the costs are shared proportionally among all the participating FIs.
Irrelevance	Ensure that participating FIs do not have an incentive either to be the first FI conducting the KYC process or to be one of those that uses the results generated by the first FI.
Privacy	Ensure that one FI cannot infer, from the system, with which other FIs a customer is working.
No-Minting	Ensure that no participating FI has an incentive to simulate having conducted the KYC process for a customer such that it can claim for compensation to which it is not entitled.

Table 1: Conditions for ensuring the viability of the system.

The suggested artifact is composed of two parts. The first part is a permissioned database that stores the documents that require a certain privacy. The second part is a distributed ledger that serves as an immutable record and clearing system via which to proportionally distribute the costs of the KYC process among the participating institutions. The system is held and managed by the regulator, who enables the database and the DLT infrastructure. This implies that the national regulator develops and maintains the fabric layer and therefore plays a central role in the system. The clearing itself, however, is conducted via the smart contract, which comes along with very low clearing costs for this solution. The artifact works as follows.

1. A number $k > 3$ of financial institutions and the national regulator agree to interact with the artifact and set the average price m of conducting a core KYC verification process. The regulator establishes a digital currency with a fixed exchange rate against the national currency. This automatically assigns value to the token used in the system. In terms of the abovementioned European Central Bank (2012) classification, this system would be framed as a Type 3 virtual currency scheme. Each financial institution can purchase digital currency in exchange for national currency, such that it can later on compensate other member financial institutions for the verifications that they conduct. The purchased digital currency can be distributed across as many different accounts as each financial institution desires. Since the system is run by the regulator, no financial institution can know to which financial institutions the other accounts belong. Only the regulator is aware, with certainty, of the activities of each financial institution.

2. Whenever customers approach a member financial institution to be validated in terms of KYC for the first time, they are granted a new account (with a public and a private key) through the systems interface. For the sake of brevity, we refer to the first financial institution that conducts the core KYC verification for a customer as the "home bank". Once customers have been granted an account in the system, they can share with the home bank their public key and the documents that must be analyzed. The exchange of these documents occurs outside of the distributed ledger to protect the privacy of the customer. The home bank will keep these documents in its local database. Once the bank decides to validate or reject a customer, it stores a digitally signed document in the smart contract of this customer and this includes the result of the core KYC verification process (*verified or rejected*). Additionally, the hash of each of the documents submitted by the customer, documents that have been used for the verification, is also stored by the home bank on the distributed ledger. Once the validation has been conducted, the home bank creates a "document package" for the customer, which contains the documents submitted by the customer and that have previously been hashed, as well as the digitally signed document that summarizes the KYC verification process and includes the result of the core KYC verification. This document package is stored in the bank's local database as well as in the permissioned database managed by the regulator. At this

stage, only the customer and the home bank have the documents package. Further, the home bank creates a smart contract for this customer, a contract that contains a list of the public keys of the wallets of the financial institutions that have checked that the status of this customer in terms of KYC has been verified and that have paid their corresponding fraction of the verification costs. We call this list the "list of onboarding institutions". At the time of its creation, when a customer only works with the home bank, the list of onboarding institutions only contains the public key of the account that the home bank has used to interact with this customer. This list can later be enlarged as the customer interacts with further institutions. We suggest that each bank uses a single, unique, one-payment-only account to interact with each customer, since this will later on protect the privacy of financial institutions and customers.

3. Whenever customers approach an institution other than the home bank with the intention of working with it, they can share with it their public key and key and the address of the original smart contract in which the home bank wrote the result of the core KYC verification process. Further, they can grant this institution access in the permissioned database to the documents package previously created by the home bank, such that it too can read them and validate the customer. Further, by reading the smart contract, the new financial institution can see how many institutions have worked with the customer so far, since it can see how many public keys appear in the list of onboarding institutions. To be added to this list, a financial institution has to pay the proportional part of the average price m of conducting a core KYC verification process. Specifically, this institution has to pay $\frac{m}{k}$ to the smart contract. Note that $k-1$ is the number of institutions that have worked with this customer so far (i.e., $k-1$ is the number of institutions that are listed in the list of onboarding institutions). The smart contract then sends the compensation that it has received, divided into equal parts between the $k-1$ institutions that had previously worked with this customer, and adds the public key of the account from which it has received the payment to the list of onboarding institutions. The payment is made in the cryptocurrency issued by the regulator

4. This mechanism ensures that all the financial institutions that work with one given customer share the costs of the core KYC verification process proportionally; that is to say, if the k-th institution that starts working with a customer always contributes with $\frac{m}{k}$ and this contribution is distributed in equal parts among the accounts of the other $k-1$ institutions, all the institutions that work with the customer end up paying the same fraction of the average price m of conducting a core KYC verification process. It is easy to see that for $k=1$ only the home bank works with the customer and that it bears the full average cost m of conducting a core KYC verification process, since no other institution is compensating it for the work conducted, which is worth m. For the case in which $k=2$, the second financial institution to join pays $\frac{m}{2}$ to the smart contract, which automatically sends this compensation to the home bank, such

that both institutions bear a cost equal to $\frac{m}{2}$. Let us assume now that this system works for a number $k \geq 2$, such that the k-th institution pays $\frac{m}{k}$. So far, each of the other $k-1$ institutions has paid $\frac{m}{k-1}$ and now receives an amount equal to $\frac{m}{k(k-1)}$ from the last institution to join. Hence, the cost for each institution equals $\frac{m}{k-1} - \frac{m}{k(k-1)} = \frac{m}{k}$.

The smart contract contains the documents' hash codes, the public key of the home bank, the certificate of approval, which conveys that the customer has been validated, and an array called "onboarded" with all the public keys of the financial institutions that have paid the proportional compensation amount to the home bank. This system ensures that the core KYC process only has to be undertaken once, by the first institution with which a customer intends to work, but that its result can be used by as many financial institutions as required by the customer.

This specific setting shows how, for a customer that works with k financial institutions, the exchange of documents and core KYC verification need only be undertaken once (and not k times as is the case in the current setting). Furthermore, the total cost of conducting the core KYC verification for one customer is now the cost m of one single KYC (and not $k \times m$, as in the current practice).

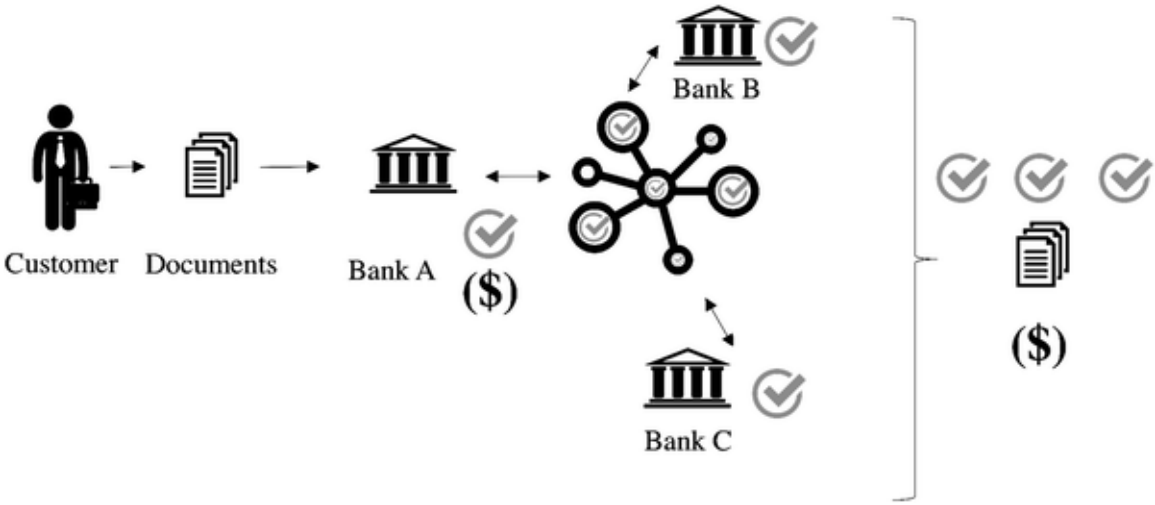

Figure 3: Proposed work flow and cost structure of KYC after the implementation of the artifact.

Figure 3 illustrates the same example case as that presented in Figure 1, but this time following the introduction of the proposed system. The system enables the same customer to work with the same three financial institutions, but now the exchange of documents and the core KYC verification process only occur once and the costs are reduced to a third.

This system fulfills the four previously defined conditions: proportionality, irrelevance, privacy, and no

minting. With regard to privacy, since each financial institution only uses one account for each customer, and it is therefore not possible to identify which institution is behind which public key, privacy, for customers and financial institutions, is ensured. Only if one customer would work with all the institutions in the system would all the institutions be able to infer that this was the case. However, since financial institutions use only one account per customer, their privacy would still be guaranteed with regard to the rest of the customers. The no-minting condition is fulfilled, since only by paying can an institution be added to the onboarding institutions list of a customer that approaches it. Since the action of compensating other institutions for the core KYC verification process that has been conducted can only be triggered by a real customer approaching an institution, no institution has an incentive to fake smart contracts claiming that it has conducted a core KYC verification process, since in such a case there would exist no genuine customer behind such a process that would subsequently approach another institution and ask to be verified.

1.6 Implementing the Redefined KYC Solution

In this section we discuss the implementation considerations of the DLT-based KYC solution previously described. It is important to note that the implementation of such a system would have significant implications for the financial sector and that it would therefore need to be carried out in close coordination with the regulator. Further, many of the dimensions of the system would depend on specific national guidelines and legislation. Hence, in this section we discuss both the suggested system and two variations on it that offer different degrees of centralization and thus make possible its implementation. We also discuss alternative designs and look into the challenges and benefits of those designs.

1.6.1 Design of a KYC Solution

The system proposed in Figure 4 explains the new KYC process using the example of a customer that approaches two financial institutions. In a first step, the customer approaches the home bank and provides the required KYC documents for verification. The home bank uses the system's application (which is installed at each of the participating documents onbanks) to handle the process of document exchange with the customer outside of the distributed ledger and to store these documents in its local database. When any document is processed by the home bank, the hash of the document is stored on the distributed ledger. Once the home bank has validated the customer, it can create the abovementioned document package, which contains all the documents that have been used (and previously hashed) to grant the verification status, as well as the digitally signed document that grants verification to this customer. Later on, the customer can provide access to this document package to any other institution with which it intends to work. Hence, the next institution that needs to validate this customer in terms of KYC can

use the local client application and communicate with the smart contract of the customer in order to obtain the customer's status, inscribe itself in the list of onboarding institutions, and handle the necessary payment over the blockchain as described in the previous section. Further, since this institution has been granted access to the document package by the customer, it can store a copy of it on its own database.

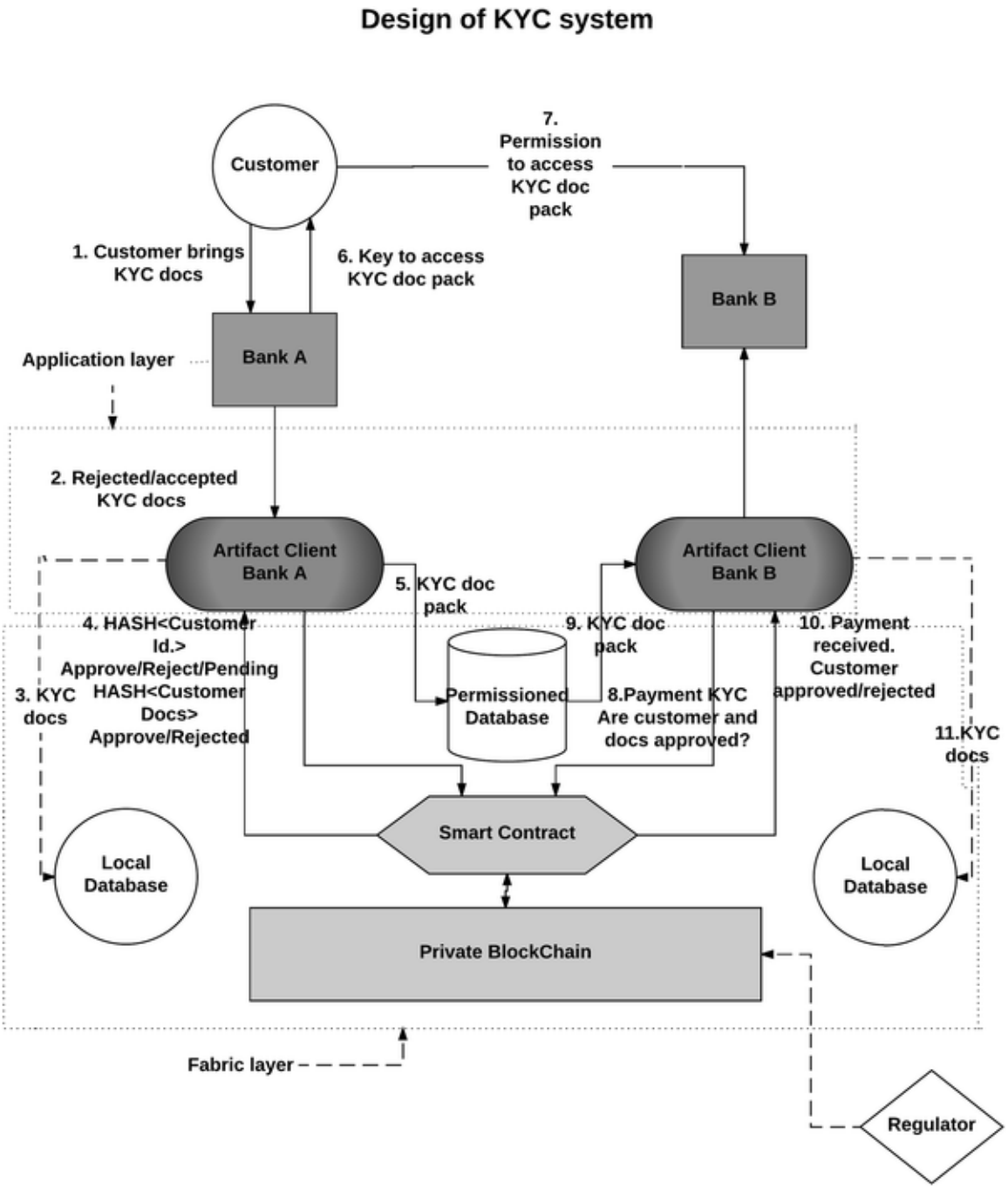

Figure 4: Design of the KYC solution.

In the proposed solution, the regulator is assigned a central role as a trusted third party (TTP) and owner of the "fabric layer". This could represent a possible shortcoming of the system if – for example – the regulator were corrupt, or compromised by hacking or by insider fraud. This is indeed an aspect

that can be further analyzed in the future. In order to mitigate this potential shortcoming to a certain extent, the TTP characteristics described by Lee et al. (2016) could be incorporated.

1.6.2 Decentralized KYC Solution

The solution proposed in the previous subsection can be further decentralized with the following modifications. First, if the DLT part of the solution were implemented directly on the Ethereum network rather than using a private blockchain, any attempt to change the information on the blockchain would be made more difficult due to the existence of a large mining community that is harder to corrupt. Second, the regulator could be removed from the system, thus precluding the risk of there being a party that has an unlimited view of the system. Last, some further efficiency could be introduced by storing the data only at the financial institution that has actually approved the customer. This solution is shown in Fig. 5. While we acknowledge these benefits, our discussions with experts indicate that in most Western countries the risk of a corrupt regulator is considered low when weighed against the benefit of the higher financial stability that would result from the regulator's ability to easily and routinely check the KYC process. Furthermore, storing the documents locally ensures that any bank that works with a client would check of the KYC documents whenever it wished. In our proposed design we have used a private distributed ledger and not a public one. This decision was based on the feedback received from the finance executives consulted during the DSR process, who stated that banks would not be comfortable having customers' private information available on a public distributed ledger (even if only hash code values of documents and the key to decrypt the customer document package were to be kept on a public ledger). This is understandable, as potential bugs in the smart contract or reverse engineering of the smart contract bytecode could lead to the risk of exposing information unintentionally. Luu et al. (2016) scan 19,366 smart contracts on Ethereum and find vulnerabilities in 8833 of them. The stated concerns of the finance executives consulted are, then, well grounded. Further, the whole compensation scheme that enables the cost reduction and cost sharing within the system is only possible thanks to the use of DLT.

A more mature DLT would allow for a ledger in which stored documents can be held completely privately. This would make possible a decentralized, permissioned database held on a blockchain. In such a system, the document package would only be stored on a distributed ledger, and not on a central database managed by the regulator. The projects R3 Corda and Hyperledger are moving in this direction. While these projects are not mature enough currently, they may well be in the near future.

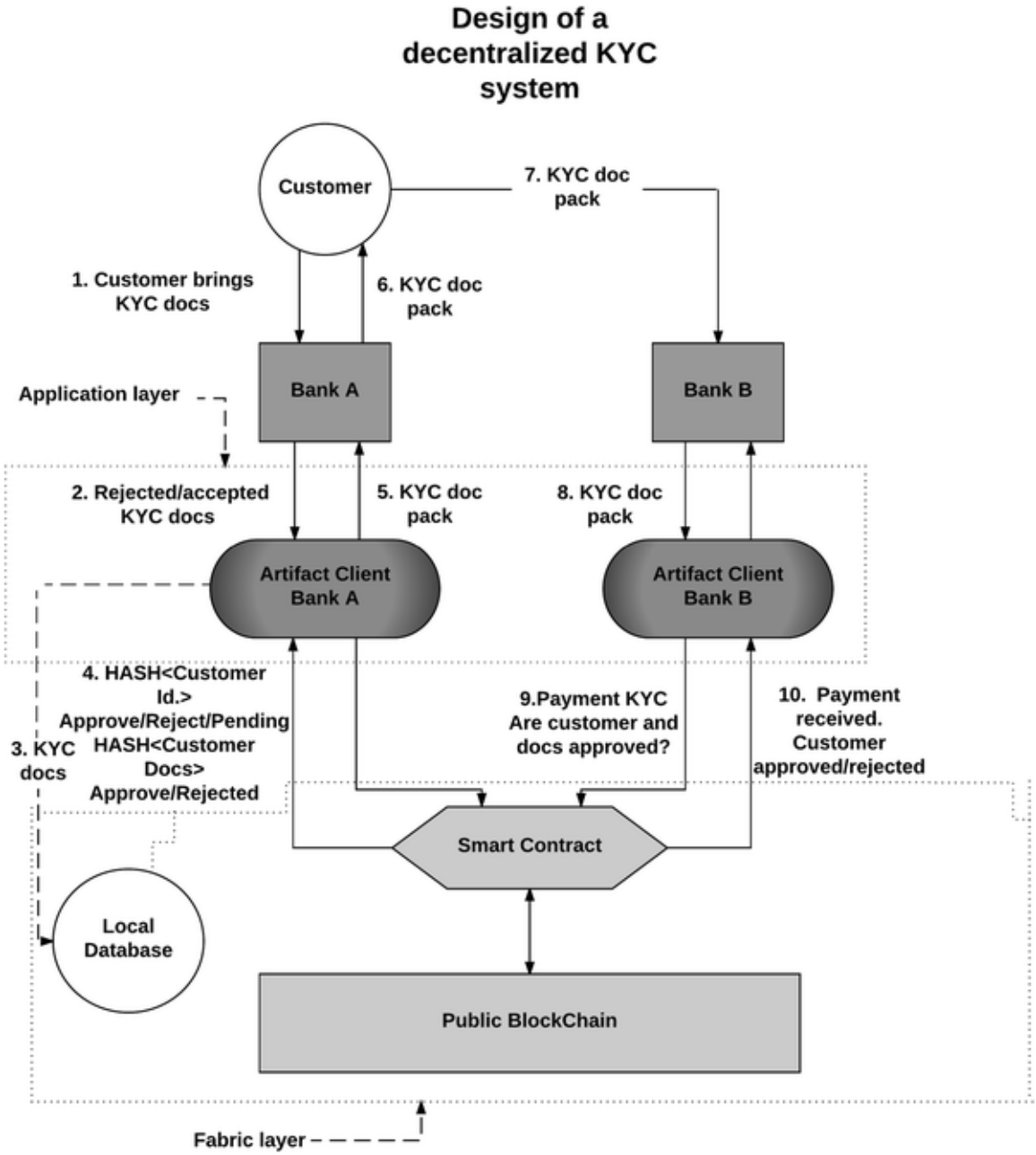

Figure 5: Design of the distributed KYC system. The blockchain is public, the documents are only kept by the home bank and the regulator does not have privileged access.

1.6.3 Centralized KYC Solution

It is possible to benefit from cost sharing during the KYC process by using a different, fully centralized KYC artifact. This would require only one party being allowed to approve or reject customers. One such centralized solution would be to transfer the entire KYC responsibility to one specialized entity or a regulator-operated KYC office. In such a design, the customer would need to be authorized by the entity and, subsequently, each bank that wanted to work with that customer would obtain a permission to do so from the centralized authority. This solution is shown in Figure 6, and while it is unlikely to be adopted

as it creates an additional cost for the regulator and in essence frees banks from the responsibility of knowing their customers, there are some significant benefits to be gained from such a solution. The main benefit is that by removing the costs of KYC from banks (and other financial institutions) we reduce significantly the cost of forming a new financial entity and, in this way, open the market up to increased competition. Furthermore, this reduction in costs for banks would lead to lower fees for customers and lower costs for doing business in a given country. That in turn would benefit a country that uses a centralized KYC solution as that country would be perceived as being open for business and competitive without necessarily compromising AML or KYC requirements.

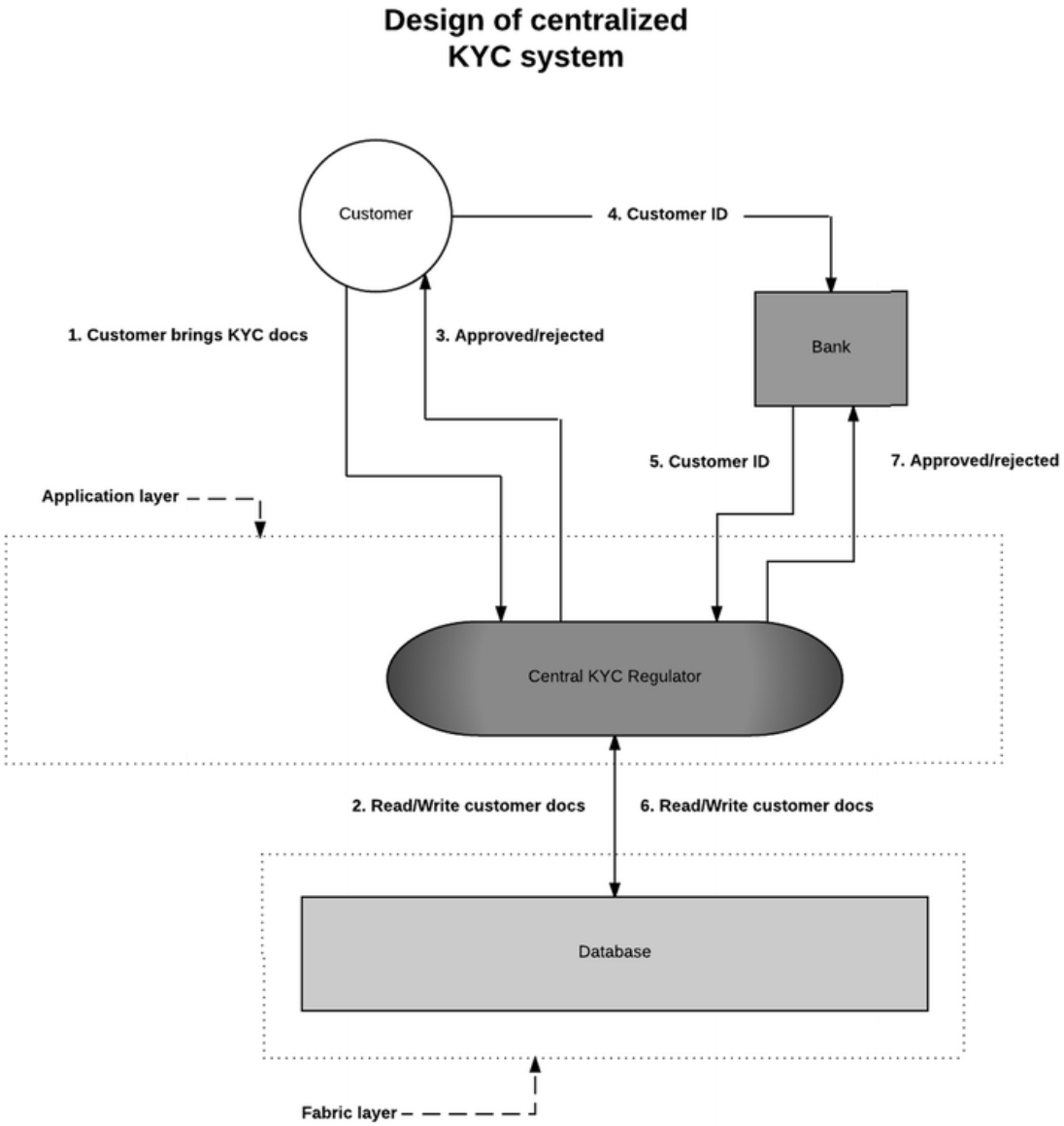

Figure 6: Design of the centralized KYC system.

1.6.4 The Use of Distributed Ledger Technology

Having presented a solution, it is worth considering why the use of DLT represents an improvement compared to other possible technologies. First, there would be improvements in terms of auditing and tracking. This is advantageous for the national regulator since it provides a clear record of the information that financial institutions verify prior to the opening of accounts, and could serve as a single point of truth should disagreement occur. And the immutable nature of the record created by DLT-based solutions cannot be matched by other technologies. Second, the proposed system allows collaboration between financial institutions that do not necessary trust one another. Specifically, given that financial institutions compete for customers' assets and accounts, only a system that allows for anonymous collaboration – such as anonymous compensation and anonymous document sharing – would gain the support of financial institutions. Third, one of the major contributions of the solution proposed here is that an institution can be anonymously and proportionately compensated by others for the efforts conducted to verify a customer. This is only possible due to the features of the distributed ledger, which allow institutions to communicate with one another without revealing their identities but ensure that each institution abides by all relevant regulations at all times. Fourth, it is important to note that the system proposed here – irrespective of the technology used to enable it – is, in essence, a system for interbank collaboration. Since financial institutions are studying broader interbank collaborations based on DLT – such as the R3 project – it seems logical to propose a system such as the one presented here, which already takes core DLT features into account, such that it can, in the future, be integrated into a broader DLT-based framework. Last, and taking into account that such a novel system would in any case need a clearing instance to settle the compensations, DLT eliminates high central authority fees. All in all, the solution proposed here from DLT for the following reasons: the application of this technology allows for the automation of a process, increases the information available if a dispute should occur, reduces settlement time compared to other technologies, and reduces business costs.

1.7 Conclusion

This paper has suggested an IS to reduce the aggregated cost of KYC in a jurisdiction by means of DLT. The main efficiency gain that this IS proposes is the avoidance of the same tasks being duplicated by different financial institutions. Additionally, this paper has shown how it is possible to distribute the costs of the core KYC verification process proportionally among those financial institutions, solutions that require the verification process be carried out for one given customer, and has defined a series of conditions that the IS in question needs to fulfill in order to ensure the correct incentive structure for the participating institutions. The maximum total cost saving per customer generated by the proposed IS can be measured as $\sum_i m_i \times (k_i - 1)$, where m_i, is the cost of conducting a full core KYC verification

for a customer i, and k_i, is the number of financial institutions that conduct business with customer i. This implies that the monetary savings brought about by the proposed IS and the increased efficiency that it would deliver for both customers and institutions are significantly affected by the number of financial institutions that participate in the system. The proposed IS has emerged from the application of design science research to the problems of high costs for financial institutions and the low satisfaction of customers when conducting a core KYC verification process. The fact that the smart contracts in which the information is stored would be owned by the customers and not by the participating institutions already addresses the paradigm shift taking place with regard to consumer data in light of the General Data Protection Regulation (GDPR), which will come into force in 2018 (European Commission 2016). For example, a simple extension of the system could oblige the client application running at each bank to regularly check in order to detect if a customer has decided to no longer work with the bank and ensure that customer's private documents are deleted. Performing a core KYC verification process on a distributed ledger has many intersections with ongoing research in the area of digital identity in distributed ledgers. One question that arises here is that of the location in which customers' sensitive documents would be stored. In the proposed IS, all the information is stored locally by each bank, as well as in a permissioned database maintained by the regulator. This is primarily due to the high cost of storage on the Ethereum platform on which the artifact was first designed. It is possible to conduct other designs based on permissioned, contractually based solutions such as R3CEV's Corda or Monetas, both of which are currently generating a lot of interest. Corda and the Ethereum blockchain have similarities, but the former is – in its essence – the combination of a distributed database and a Java Virtual Machine, enabling parties on the network to execute bilateral transactions involving sensitive information that is not revealed to the public. These kinds of solutions could offer new approaches to providing distributed but private document exchange between customers and financial institutions that include storage possibilities for larger documents. However, solutions such as Corda are still in their early stages of development and privacy with regard to the customer data that is shared in such a system is a concern that needs to be thoroughly addressed.

Regardless of the chosen approach to using DLT, be it a distributed database or a private, restricted, or public blockchain, our research suggests many opportunities to increase efficiency in the financial system. More specifically, a significant reduction in costs for the participating institutions and an improved experience for customers could both be delivered by such a system. Furthermore, the system would – thanks to the decreased regulatory costs of KYC – lower the barriers to operating a financial institution, thus opening the financial market up to further competition.

Essay 2

Optimised and Dynamic KYC System Based on Blockchain Technology

Optimised and Dynamic KYC System Based on Blockchain Technology

José Parra-Moyano
University of Zurich
Switzerland
jose.parramoyano@uzh.ch

Tryggvi Thoroddsen
University of Zurich
Switzerland
tryggvi.thoroddsen@uzh.ch

Omri Ross
University of Copenhagen
Denmark
omri@di.ku.dk

September 2018

Abstract

Systems that use blockchain technology to improve the know-your-customer (KYC) process have only been proposed at a conceptual level and all share certain attributes that make their adoption by financial institutions (FIs) very difficult. We propose and program a blockchain-based system that reduces and shares out among the financial institutions that work with a customer the costs of the KYC process and also makes it possible for FIs to dynamically update information related to customers and disseminates this information among participating FIs. Additionally, and by means of a distributed database architecture, our system addresses some of the attributes that hinder the adoption of previously proposed solutions by FIs. The result is a programmed, stand-alone solution that can be implemented by FIs to reduce the cost of the KYC process without requiring any central instance to store the customer's data, and in which FIs share the initial costs of the KYC process as well as the running costs of keeping the information about customers up to date. Our system increases the levels of security and regulatory compliance in the KYC process and significantly reduces the cost of that process for all parties involved.

Keywords: Blockchain, Information Systems, KYC, DLT, Distributed Databases

Note: A version of this paper has been published as Parra-Moyano, J., Thoroddsen, T., and Ross, O., Optimised and Dynamic KYC System Based on Blockchain Technology, Int. J. Blockchains and Cryptocurrencies, 1 (1), 85-106, 2019.

2.1 Introduction

The know-your-customer (KYC) process that financial institutions (FIs) are obliged to follow whenever they establish a financial relationship with a new customer represents a significant financial burden for FIs but creates no productive added value. The KYC process is made up of a series of routine tasks that, when carried out, are meant to verify the lawfulness of a potential customer's activities. Every FI needs to follow the KYC process before even starting to work with a new customer. The cost of KYC is rising. Thompson Reuters (2017) estimates that on average large financial institutions with turnovers in excess of USD 10 billion increased their annual spending related to KYC obligations from USD 142 million to USD 150 million during 2016. The same report contains the prediction that spending on KYC-related tasks would increase by 11 percent over the 12 months following its publication. According to Thompson Reuters (2017), corporate customers work on average with 11 FIs, which implies that this—costly—KYC process is repeated on average eleven times for each corporate customer. The average time an FI takes to "onboard" a corporate customer is 26 working days.

The increasingly widespread use of blockchain technology has led to the development of new systems that are meant to improve the efficiency of the KYC process and to enable cooperation among FIs. If achieved, both these goals will lower the costs of KYC. Parra-Moyano and Ross (2017) were the first to suggest that the KYC process should be conducted only by the first FI that wishes to work with a given customer, and that the result of conducting the process (proof of the lawfulness of that customer's activities and of the customer's "validation") should be shared in an anonymized and secure form with all FIs that subsequently wish to establish a financial relationship with that customer. The system proposed by Parra-Moyano and Ross (2017) also includes a structure that distributes, proportionately, those KYC costs initially borne by the first FI in order to work with a given customer among all the FIs that subsequently work with that customer, including that first FI. While their proposals constitute an innovative approach to reducing the costs of KYC, they involve certain inherent inefficiencies that make their implementation in a corporate environment difficult. These include the need for a trusted third party (TTP) to store the customers' data and carry out the financial compensations between FIs and that no updates or changes in the information status of a customer are possible. Parra-Moyano and Ross (2017) also only develop their proposed system on the conceptual level.

Our aims with the present paper are to review the work carried out by Parra-Moyano and Ross (2017) as well as a range of the other blockchain-based systems thus far proposed as ways of improving the KYC process and to tackle the open issues that make the implementation of these systems in the corporate

environment difficult. This enables us to suggest a system that can be realistically implemented in the financial sector and to develop a stand-alone proof of concept (PoC) of the system that can be used as a foundation from which corporations and regulators will be able to explore and—eventually—conduct the implementation of blockchain-based KYC solutions. In Section 2.2 we briefly describe the current KYC process and the requirements it must fulfill. In Section 2.3 we provide a brief introduction to blockchain technology and innovations in the architecture of distributed databases, focusing on the attributes of these two technologies that make the system that we propose possible. In Section 2.4 we analyze those systems that claim to improve—by using blockchain technology—KYC as it currently stands. In Section 2.5 we describe how we used design science research (DSR) to refine these previously proposed systems and to derive our PoC. In Section 2.6 we describe, from a non-technical perspective, the refined system that we propose, and we present the code that yields the PoC. In Section 2.7 we conclude.

2.2 Current KYC System

In recent years, the KYC due diligence process has evolved from a simple formality into a thorough process supervised by national institutions. The Financial Action Task Force (FATF)—an intergovernmental body established to combat money laundering and the funding of terrorism—sets the international standard for KYC. That standard is outlined in the FATF Recommendations (The Financial Action Task Force, 2012-2017), a document that was first published in 2012 and was updated in November 2017. We can paraphrase the FATF Recommendations' minimum requirements for FIs conducting the KYC process as follows:

1) Identify the customer and verify that customer's identity using reliable, independent source documents, data, or information.

2) Identify the "beneficial owner", verify the beneficial owner's identity, and understand the ownership and control structure of the customer.

3) Understand and obtain information on the purpose and intended nature of the business relationship.

4) Conduct ongoing due diligence on the business relationship throughout the course of the relationship to ensure that the transactions being conducted are consistent with the FI's knowledge of the customer.

The first three of these requirements must be met by each FI before it establishes a financial rela-

tionship with a new customer. Thus, if one customer works (or intends to work) simultaneously with n FIs, the KYC process for that customer will be repeated n times. Although each FI is responsible for its own KYC process and must conduct due diligence independently of other FIs, a core portion of KYC due diligence—namely, points 1, 2, and 4 in the above list, is a routine process that is carried out in parallel by all FIs that work (or intend to work) with the same customer. Thus, costly tasks are carried out repeatedly and in parallel whenever a customer works with two or more FIs. Figure 7, which we derive from Parra-Moyano and Ross 2018, schematically describes the current scenario.

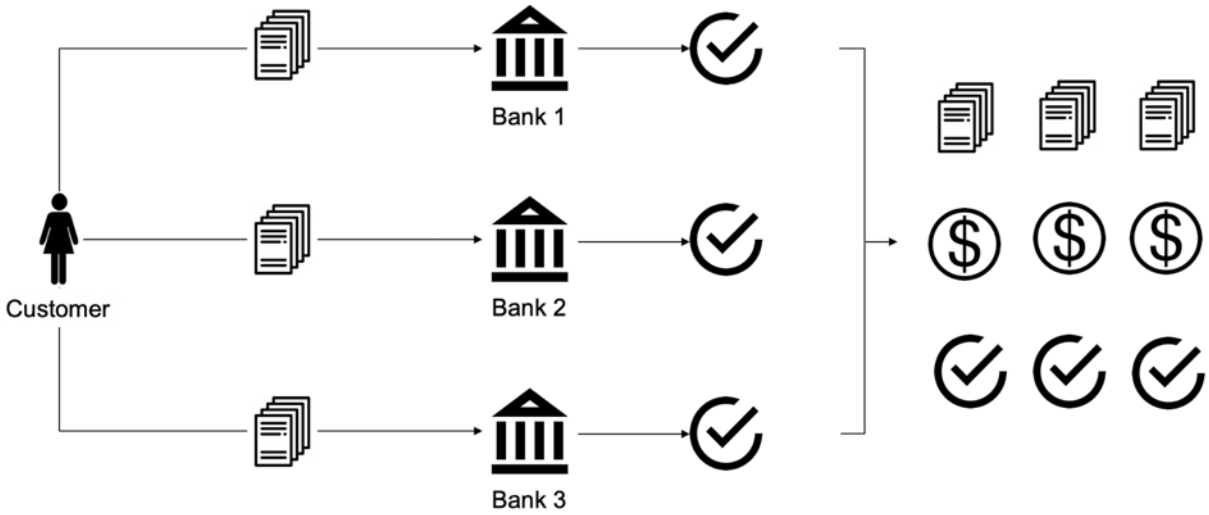

Figure 7: Current KYC Scenario, (adapted from Parra-Moyano and Ross, 2017).

2.3 Blockchain and Distributed Database Technology

This section briefly introduces the two technologies that we use as basis for the solution that we propose in Section 2.5—namely, blockchain technology, which was introduced by Nakamoto (2008), and the private distribution of data across distributed databases, introduced by Siegenthaler and Birman (2009a and 2009b).

2.3.1 Blockchain Technology

Blockchain technology offers a global, distributed transactional database in which nodes are linked to one another by a peer-to-peer (P2P) communication network with an own layer of protocol messages for node communication. Users of a blockchain can reference one another using their respective public keys and can use their private keys to cryptographically sign messages and transactions (Glaser, 2017). Although blockchain technology has gained notoriety primarily due to the advent of crypto-currencies, such as Bitcoin, which was introduced by Nakamoto (2008), researchers and practitioners are applying it

in a number of ways to improve existing information systems (IS) and make them more efficient.

One of the ways in which blockchain technology contributes to the improvement of IS is that it makes the execution of "smart contracts" by any node that has access to the blockchain possible. Smart contracts are computer protocols that facilitate, verify, or enforce predefined clauses whenever a given set of conditions is met (Parra-Moyano and Ross, 2017). Once a smart contract is triggered, it can carry out automatized, predefined actions. Currently, the three largest distributed ledger platforms that facilitate smart contracts are Ethereum, from the Ethereum Foundation; Hyperledger, from IBM; and Corda from R3. Because, in a blockchain, a copy of the ledger is distributed to each node, there is no need for a TTP to act as a notary with regard to processes that involve participating nodes that do not trust one another.

The validity of the information stored on a blockchain's ledgers is ensured by the network's nodes with the help of a secure hash algorithm (SHA). Blockchain technology uses an SHA to translate the contents of a block into a cryptographic fingerprint referred to as a "hash". An SHA can also be used to generate from a digital document a unique "fingerprint" of that document, such that this fingerprint cannot be replicated unless it is generated from the exact same document. This ensures that all of a blockchain's participants can easily verify the authenticity of any document previously hashed simply by hashing it again and comparing the hash they generate to the hash that was previously generated using the authentic document. Further, the hash does not reveal any information about the contents of a document, just as analyzing a human fingerprint can help one to prove the identity of an individual but fails to reveal—for example—the features of that individual's face. In a distributed ledger with multiple nodes, the information recorded by the network is stored sequentially in a list of records that is divided into blocks and distributed to all nodes on the network. The information in each individual block is then used by the system's protocol to generate a secure hash that identifies that specific block. Each subsequent block records the hash of the previous block such that all blocks are chained together sequentially making it impossible to change information in one block without changing all previous blocks. If one node alters the information on its ledger and tries to interact with the network using what is, thus, "false" information, the hash will no longer match the ledger distributed to the other nodes on the network and the transactions that this node attempts to conduct will not be accepted by these other nodes. The process of verifying transactions and ensuring that blocks have not been altered is carried out by the nodes of the network.

Blockchains are, most commonly, either "public" or "permissioned". A permissioned blockchain limits the number of nodes that can access it or that can approve the hashes that are to be saved on the ledger. A public blockchain, meanwhile, has an unlimited number of nodes and is accessible to all.

In order for it to be maintained, a blockchain requires a protocol that defines the roles and rules that apply on it. The many protocols that can be implemented in blockchain technology include proof of work (PoW), proof of stake (PoS), and proof of authority (PoA). PoW incentivizes the participating nodes to spend computational power (work) and write new blocks. The fact that spending computational power is costly means that rewriting the blockchain is expensive; this secures the blockchain against fraudulent attacks, which indeed need to rewrite the blockchain in order to be successful. To compensate "miners" (nodes that verify transactions in all kinds of mineable blockchains), the protocol provides a reward in the form of crypto-currency to the first miner that writes a valid block. The PoS protocol relies on a smart contract that holds deposits—of a crypto-currency—made by nodes that wish to act as miners. The node that supplies the largest amount of crypto-currency is assigned the authority to mine by the blockchain's owner. Once a node has been granted this authority it no longer needs to rely on computational power to be allowed to mine. The PoA protocol defines that pre-authorized nodes act as miners and add blocks to the blockchain. Instead of using hash power to write valid blocks—as is the case with PoW—or providing funds in order to be granted the right to mine, PoA nodes are able to add blocks to the blockchain at any time.

2.3.2 Private Information Sharing Across Distributed Databases

Siegenthaler and Birman (2009a and 2009b) introduce a database architecture that allows the electronic sharing of privacy-sensitive data across distinct nodes and respects very high privacy standards. This architecture was constructed to allow hospitals—holders of patients' medical data—to share patient data with each other to improve patients' treatment and respects three privacy principles that ensure that only the necessary information about patients is shared and that hospitals making and receiving queries do not learn anything about one another that can reveal information about the patients that is sensitive or is irrelevant to a patient's treatment. Specifically, the database architecture that they introduce allows entities to store different pieces of data such that the following three principles are respected:

1) Data privacy. Queriers learn only the answer to their query, not any of the data used to compute that answer.

2) Query privacy. The data owner does not learn the particulars of the query, only that a query was performed against a particular patient's information.

3) Anonymous communication. Neither queriers nor data owners know who the opposite party is. For this architecture to work, the database schemas of the hospitals are irrelevant; it is sufficient for the data

producers to provide a read-only API to the members of the network. This system allows data to be shared between nodes in a secure and encrypted manner.This system does, however, require a TTP to manage both access and the right to perform queries against other nodes' databases.

For this architecture to work, the databases' schema of the hospitals is irrelevant; it is sufficient for the data producers to provide a read-only API to the members of the network. This system allows the data sharing between nodes in a secure and encrypted manner. This system does, however, require a TTP to manage both access and the right to perform queries against other nodes' databases.

2.4 Previously Proposed KYC Systems Based on Blockchain Technology

Since the KYC process—whether observed from a national or an international perspective—is characterized by many duplicated tasks carried out by agents that do not trust one another, it seems that blockchain technology may have significant promise when it comes to reducing inefficiencies and costs and to offering a more efficient structure under which to conduct KYC. It comes as little surprise then that a number of bodies and organizations have suggested various approaches to using distributed ledger technology (DLT) to improve KYC systems.

Parra-Moyano and Ross (2017) propose a blockchain-based system that improves the efficiency and reduces the cost of the KYC customer-onboarding process. Their system is meant to be run by a national regulator, which provides and maintains the system's physical and operational structure. In this system, the KYC process is carried out only once, by the first FI that is approached by a customer. When that customer approaches another FI with the aim of establishing a financial relationship, this second FI can see—by consulting the blockchain—that the KYC process has already been carried out (in this case by the first FI) and can thus focus solely on certain, limited aspects of KYC (namely, understanding the customer's activities) and does not need to perform routine, mechanical document verification. Further, the system includes a mechanism that distributes the costs of conducting the KYC process proportionally among all participating FIs that work with a given customer. While this is the most comprehensive work on blockchain-based KYC systems that the literature currently contains, it possesses a series of characteristics that hinder its implementation in the corporate environment and therefore needs to be improved upon if it is to become truly useful for FIs and regulators. The first aspect that needs to be improved is the fact that in Parra-Moyano and Ross' (2017) solution the TTP must periodically check that the FIs in the system have paid the proportion of the cost that they are meant to have paid and have not simply used the system to verify that the documents presented to them by any given customer have previously been validated. The second aspect that requires improvement is the fact that in the system

proposed by Parra-Moyano and Ross (2017) the status of a customer cannot be updated by an FI in a decentralized way; rather, the KYC onboarding process is only conducted once for each customer and by one FI only, and the system does not envisage the potential need for periodic updates with regard to a customer. Thus, if a customer is validated by the first FI but its status later needs to be changed due to—for example—irregularities in its activities, information with regard to the customer's new status cannot be disseminated using the system to all those FIs that work with that customer. A third aspect that is susceptible to improvement is that the proposed system is unable to make dynamic compensations between participating FIs. Such a dynamic compensation system is required in order to allow for financial compensations among the FIs participating in the system over time, specifically whenever an FI needs to update a customer's KYC status or history. A fourth aspect that could be improved concerns the storage of customers' documents. Parra-Moyano and Ross (2017) propose a complex database architecture in which customers need to store these data privately and circulate them among the FIs with which they want to work. Such a structure is costly, and it becomes clear—when one compares the customer journey that emerges from this structure with the existing customer journey—that the self-storage aspect would be a disadvantage.

The R3 Project, run by a consortium of banks, conducts applied research on blockchain applications. R3 runs Corda, an open-source distributed ledger platform designed to record, manage, and automate legal agreements between businesses. The Corda network is made up of nodes, where each node represents a run-time environment hosting Corda services and executing applications, or "CorDapps". CorDapps are participant applications that execute contract code and communicate using a flow framework to achieve consensus over some given business activity. While there already exist CorDapps for asset trading (IRS Demo and Trader Demo), as well as for portfolio valuation (see SIMM and Portfolio Demo—also known as the Initial Margin Agreement Demo), there does not yet exist a functioning CorDapp for KYC/AML. Rutter (2018) describes the benefits of decentralizing the KYC process and provides a conceptual description and comparison of two different decentralized scenarios run on Corda—namely, the "Self-Sovereign Model" and the "Bank Sharing Model". In the self-sovereign model corporate customers create and manage their own identities and relevant documentation, granting permission to multiple participants to access this data whenever they require it. In such a system, the relationship remains one of customer to bank, with the rights and responsibilities of each laid out in a contract and the bank not necessarily storing any of the customer's data. Instead, the customer permits the release of their data to each individual bank.

The use of blockchain technology in the development of digital identity has also proved promising. Blockchain technology can be used to register and store the credentials and ID-related information of users and can act as a TTP, verifying users' identities (Shocard, 2017; and Civic, 2017). These types of

digital identity blockchain solutions may have important implications for the improvement of the KYC process.

Britton (2016) briefly analyzes the potential uses of blockchain technology in the context of KYC/anti-money laundering (AML) measures. While he considers that the KYC framework is probably one of the most suitable for the application of blockchain technology, he also addresses the difficulties of realistically establishing such a system. Specifically, he states that while there is immense potential in the application of blockchain technology for KYC, there are certain challenges that need to be addressed if one is to create a viable proposition that can be adopted by the industry. He pays special attention to the network effect that must be present in a valid blockchain-based KYC solution and claims that it could only result from collaboration among market participants working toward a mutually beneficial solution that would enable them all to focus on the customer.

The Hong Kong Monetary Authority (2016) has studied the potential benefits of blockchain technology for the financial sector. The authors of the study conclude that the technology offers the potential for banks to share identity information in an effective and secure manner, such that digitized customer records and documents could be shared among banks using a blockchain-based platform. The Authority specifically states that such an arrangement would offer a number of benefits. First, customers would no longer need to repeat the same processes and submit the same personal information to different banks for KYC purposes; second, the costs incurred due to and the resources needed for the identity-verification process would both decrease because the information in question would be readily accessible and shared via the blockchain; third, the checking of customers' history could be carried out efficiently by participating banks because customers' information would be available in the blockchain; and fourth, a better customer experience would result. The authors of the study also state that existing KYC requirements and customer-authentication processes are manually intensive and require significant resources from FIs that seek to be compliant. The authors do not, however, propose a specific design for such a system.

2.5 Research Methodology: Design Science Research

The aim of design science research is to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts (Hevner, 2004). Since in this paper we aim to propose an optimized, blockchain-technology-based KYC artifact that enhances the capabilities of FIs, design science research is an appropriate method for our purpose. Well-founded design science research, according to Hevner (2004), follows seven guidelines and results in a technology-based solution that solves a relevant business problem. These guidelines are depicted in Table 2.

Guideline	Description
Design as Artifact	Design science research must produce a viable artifact in the form of a construct, a model, a method, or an instantiation.
Problem Relevance	The objective of design science research is to develop technology-based solutions to important and relevant business problems.
Design Evaluation	The utility, quality, and efficacy of a design artifact must be rigorously demonstrated via well-executed evaluation methods.
Research Contributions	Effective design science research must provide clear and verifiable contributions in the areas of the design artifact, design foundations, and/or design methodologies.
Research Rigor	Design science research relies upon the application of rigorous methods in both the construction and the evaluation of the design artifact.
Design as a Search Process	The search for an effective artifact requires using available means to reach desired ends while satisfying laws in the problem environment.
Communication of Research	Design science research must be presented effectively both to technology-oriented and to management-oriented audiences.

Table 2: Guidelies for DSR (adapted from Hevner, 2004).

We rigorously follow these seven guidelines in order to propose an optimized KYC system. Because in this paper we propose a viable, technology-based artifact in the form of an instantiation that solves the problem of the high cost of KYC we follow guidelines one and two. We demonstrate the utility, quality, and efficacy of the artifact by following a well-executed process—that proposed by Peffers, Tuunanen, Rothenberger, and Chatterjee (2007), which is similar to the procedure followed by Parra-Moyano and Ross (2017)— and thus follow guideline three. Specifically, the process is divided into five steps, as illustrated in Figure 8: identifying the problem, defining the objectives, designing and refining the artifact, demonstrating the artifact, and evaluating the artifact. We recursively repeated the last three steps of the process to carry out a vigorous evaluation of the artifact's design.

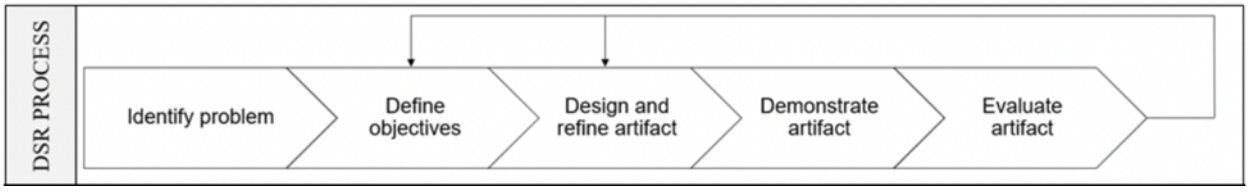

Figure 8: DSR Process (adapted from Pfeffers et al., 2007).

In following guideline four, we provide not only a blockchain-technology-based KYC solution that solves the open issues of all the previously published blockchain-technology-based KYC solutions, but also a programmed, functioning artifact that can serve as the foundation upon which other researchers and practitioners may develop the artifact further. Programing such an artifact has never been done before. To ensure our adherence to guidelines five and six we collaborated for seven months with blockchain and KYC experts from Origo, Iceland's leading systems integrator and managed service provider. Origo's experts not only systematically evaluated the progress made by our research, they also contributed—with valuable insights—to the design and the coding of the artifact. We communicate our results in a manner that is in line with prevailing academic style and language, but one that also renders the proposed solution accessible to practitioners, as is dictated by guideline seven.

Previous research has identified the problem of the increase in KYC costs for FIs, as we have explained in the previous section. But while researching those systems that have been proposed explicitly to improve the KYC process by applying blockchain technology we failed to identify even one single academic publication that addressed both the theoretical conception of a blockchain-based KYC solution and its technical development in the form of a PoC. Further, during our research we identified a series of inefficiencies in the previously proposed conceptual solutions. We embodied all these problems in the definition of the following research question:

"*Can we conceptually develop a blockchain-based solution that improves the KYC process for financial institutions and that solves the open issues of previously suggested systems, while programming it and converting it into a functioning, verifiable, replicable instantiation?*"

With the problem(s) identified and our objective—embodied in our research question— defined, we recursively undertook the three remaining steps of the process proposed by Peffers et al. (2007). The first sequence of this design and refine, demonstrate, and evaluate phase was carried out in collaboration with the experts from Origo, who assessed the conceptual solution and the artifact's smart contracts, constructively criticizing those characteristics of the solution that would hinder the system's implementation in a corporate environment. Their contributions proved essential. Thanks to their corporate experience we were able to identify elements of the initial solution that required improvement if the solution was to be implementable. We refined the solution in further loops. The focus of one of these refinement loops was the dynamic compensation system that we propose here. To our knowledge, all previous blockchain-based KYC solutions propose only static compensation between participating FIs, neglecting the fact that update costs might be incurred over time due to the demands of continued compliance with KYC

regulations. Therefore, in the aforementioned loop we focused solely on establishing a smart contract that could accommodate dynamic compensation among participating FIs, and developed a solution that both allows for dynamic payments and respects proportionality in the payments made among FIs. We conducted another refinement loop focusing on the role of the TTP, a role that is central to all the solutions published thus far. We concentrated on reducing the reliance of the system on controls conducted by the TTP, ensuring instead intrinsic incentives for participating FIs. Our aim in suggesting this improvement is to increase the autonomy of the system by transforming the artificial incentives present in previously proposed systems into intrinsic incentives and to ease the implementation of the system by means of distributed a database architecture like the one proposed by Siegenthaler and Birman (2009a and 2009b), which is used in the healthcare sector to securely share patients' private information among hospitals. Combining blockchain technology and the distributed database architecture suggested by Siegenthaler and Birman (2009a and 2009b), in general and in this particular manner, has already been suggested by Parra-Moyano and Schmedders (2018).

2.6 Optimized, Dynamic KYC System

In this section we describe the process and logic of the optimized, dynamic system we propose for reducing the cost and proportional cost share of KYC. We start with a description of the assumptions and conditions that must be fulfilled by the system and continue with a non-technical description of the system. We conclude with a description of the proposed system's technical aspects.

2.6.1 Assumptions and Conditions

The solution that we propose in this section combines elements of the proposals that we describe in Sections 2.3 and 2.4, but specifically takes as its basis the system suggested by Parra-Moyano and Ross (2017) and combines it with the distributed database architecture suggested by Siegenthaler and Birman (2009a and 2009b). The four key assumptions of Parra-Moyano and Ross (2017) are:

1) All FIs with access to the system respect and work in the same regulatory framework.

2) All customers can be categorized by the effort required of an FI to conduct the KYC process for them.

3) All FIs with access to the system agree on an average cost of conducting the KYC process per category of customer.

4) A TTP, such as a regulator, maintains the system and approves the FIs that have access to the system.

Assumptions one and four are required in order to achieve the goal of proportional cost sharing among participating FIs working in the same regulatory framework. In the present paper we specify assumptions two and three in greater detail such that a system that fulfills these assumptions is able to compute not only the average cost—per category of customer—of conducting the KYC process once, but also the cost of updating, where necessary, the KYC-related information of any existing customer. Specifically, we suggest using measurable parameters that are derived from the time spent on conducting the KYC process (e.g., the size of a corporation, number of documents required, and number of beneficiaries) and that can be used to dynamically determine both these sets of costs. We state the revised assumptions two and three as follows:

2') All customers can be categorized by the effort spent on the KYC process, and by the effort required to update their KYC status.

3') All FIs with access to the system agree on the cost of conducting the KYC process per category of customer, and on the cost of updating the KYC status of a customer.

The greater detail contained in these two revised conditions implies a significant improvement on the system proposed by Parra-Moyano and Ross (2017) since it enables us to replicate, in a closer manner than proposed by Parra-Moyano and Ross (2017), the current nature and requirements of the FATF Recommendations (2017). More specifically, the fact that we allow for the dynamic but transparent measurement of KYC costs and— more importantly—for the dynamic correction and updating of the KYC status of any existing customer enables us to propose a less rigid system than that put forward by Parra-Moyano and Ross (2017).

Parra-Moyano and Ross (2017), further, define four conditions that the system must fulfill in order to ensure a correct incentive structure. These conditions are:

Proportionality: Ensure that the costs are shared proportionally among all the participating FIs.

Irrelevance: Ensure that participating FIs do not have an incentive either to be the first FI conducting the KYC process or to be one of those that use the results generated by the first FI.

Privacy: Ensure that one FI cannot infer, from the system, with which other FIs a customer is working.

No-minting: Ensure that no participating FI has an incentive to simulate having conducted the KYC process for a customer such that it can claim for compensation to which it is not entitled.

These assumptions and conditions constitute the yardstick against which we measure the applicability of the system that we propose in the following subsections.

2.6.2 Non-Technical Description of the Optimized, Dynamic KYC Process

A consortium of FIs can use their existing database architecture to construct a system like the one proposed by Siegenthaler and Birman (2009a and 2009b). In order to grant read and write permission to FIs other than the one that conducted the KYC process, we use a private, PoA-based blockchain in which only FIs belonging to the consortium can participate. These two pieces of technology (the distributed database architecture to share sensitive data and the blockchain technology to manage permissions) constitute the main innovation of our system.

Whenever a customer has yet to be registered on the network, the first FI to onboard that customer using the network (hereafter referred to as the "Home Bank") must proceed in the following manner:

1) The Home Bank gathers all necessary documents, verifies the customer's identity, and generates a digitally signed document indicating the outcome of the core KYC process—this outcome can be either "approved" or "rejected".

2) The Home Bank stores all the documents used in the KYC process as a "document package" in its own encrypted database.

3) The Home Bank creates a smart contract for the customer and on it stores the hash of the document package, the network address of the customer, and the monetary value m that corresponds to the cost of conducting the KYC process for this customer (the last of these according to the predefined category to which this customer belongs in terms of the effort required to conduct the KYC process). When the smart contract is deployed by the Home Bank, its ownership passes from the Bank to the customer.

The smart contract records the address of the Home Bank in a list referred to as the list of onboarding institutions. At this point, the only FI on that list is the Home Bank. When the customer seeks to work

with a second FI, this second FI—"Bank B" for brevity—must proceed in the following manner:

1) Bank B activates the smart contract of the customer and thus learns the proportion of the already incurred cost that it has to pay to the smart contract in order to get permission to read the documents related to the customer stored in the Home Bank's database. Since Bank B is the second FI that intends to work with the customer, it should pay the existing cost m divided by two.

2) Since the customer is willing to work with Bank B, she will grant permission to Bank B to pay the fraction $\frac{m}{2}$ to the smart contract such that once Bank B conducts the payment it automatically gets the right to read this customer's document package as stored in the Home Bank's database.

3) Bank B can now make an API call to the database of the Home Bank and read the documents regarding this particular customer using an architecture like the one proposed by Siegenthaler and Birman (2009a and 2009b).

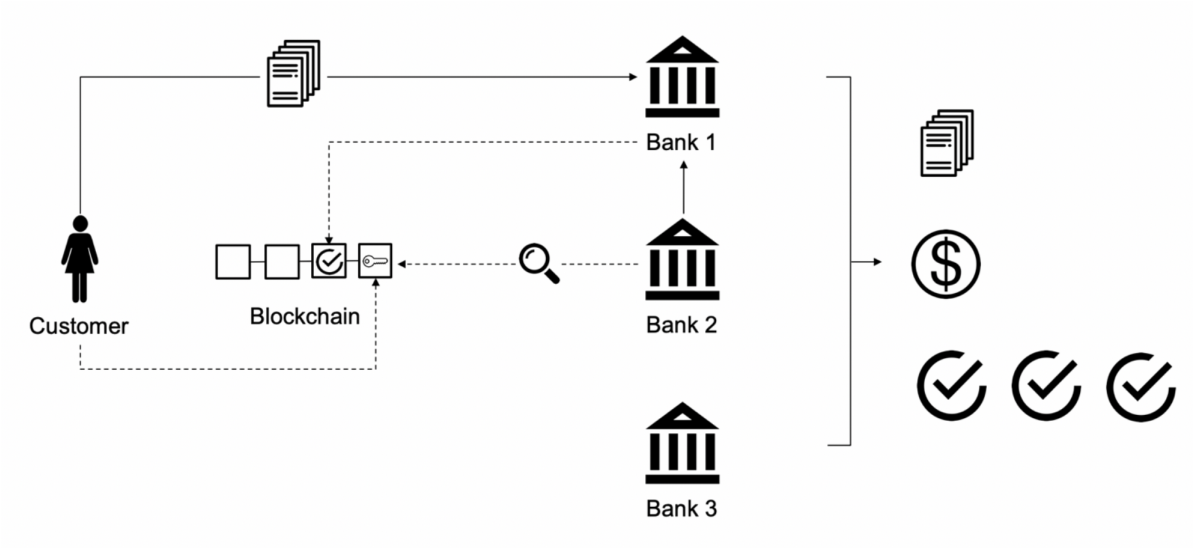

Figure 9: Schematic Representation of our System.

One of the difficulties in applying the system put forward by Parra-Moyano and Ross (2017) is caused by the fact that everything written on a distributed ledger is visible to all those who participate in the network. This visibility introduces a vulnerability to the system's incentive structure because the hashes stored on the ledger can be obtained by a user with enough technical knowledge, and that user is then able to see the KYC status of a customer ("approved" or "rejected") without having paid the corresponding contribution. In order to solve this issue, Parra-Moyano and Ross (2017) propose that all the FIs that work with a customer must pay the corresponding portion of the KYC process to a smart contract in order to appear in the list of onboarding institutions. This setup ensures that the TTP can always

conduct controls in order to verify that all necessary contributions have been paid by all those FIs that are benefiting from a KYC process or update carried out by another FI. This is, however, an artificial and not an intrinsic incentive for participating FIs: they could choose not to pay the contribution if they know that the TTP might not control all transactions for all customers. In order to transform this artificial incentive into an intrinsic one, we make use of a distributed database architecture, which allows us to eliminate the TTP and ensure that only FIs that have paid their proportion of the cost can actually read information regarding the customer.

Respecting the contribution structure derived by Parra-Moyano and Ross (2017), the system we propose here would allow all the FIs working with a customer to proportionally share the cost of the onboarding process because they would all pay $\frac{m}{k} - m$, being m the cost of onboarding process and k the number of FIs working with the customer.

Once a second institution—Bank B, in this case—has followed the steps necessary to verify that a customer has already been onboarded, two possible situations might arise. Either no update or correction to the document package of this customer is required (as suggested by Parra-Moyano and Ross (2017)) or—and this is the more likely case—a KYC update process must be followed by Bank B in order to comply with the national regulatory regime. If no update is required, the payment structure remains as presented so far. If Bank B must pay the contribution $\frac{m}{2}$ to the Home Bank via the smart contract and at the same time realizes—after checking the document package—that an update is required, the Home Bank (and, later in the process when multiple FIs have worked with the customer, all those FIs that have previously worked with this customer) must somehow be informed of this and must also proportionally contribute to the cost of the update process—a process that Bank B will carry out and from which the Home Bank is (and later, all relevant FIs are) going to benefit. In order to allow for the sharing of updated information and to enable a dynamic but private communication and compensation channel connecting FIs, we define a further variable—namely, c—which represents the "Cost of Update". We assume that $c < m$, because when following the update process Bank B already possesses the previously available information and does not need to start from scratch. Nevertheless, c, can be subject to specific factors and conditions. Were this the case, the situation would then look as follows:

1) Bank B would have to pay the contribution $\frac{m}{2}$ to the Home Bank via the smart contract and would also realize that an update process is required in order for it to comply with the regulatory regime.

2) Bank B would follow the update process and would generate a digitally signed document that indicates the process' outcome, which again can be either "approved" or "rejected".

3) Bank B would store all the documents used in the KYC process as an "updated document package" in its own database. This updated document package would now be readable by all those FIs that are listed in the smart contract and that have, directly or indirectly, paid for the update (at this stage, Bank B and the Home Bank).

4) Since Bank B has carried out the update process and borne its costs, the appropriate proportions of these costs must be paid by the institution(s) listed in the list of onboarding institutions (at this stage only the Home Bank) to Bank B.

In order to ensure that step four is respected, we suggest a time lag (or block sequence that needs to be added to the blockchain) between the moment at which an FI, k, other than the Home Bank pays its contribution $\frac{m}{k}$, via the smart contract and the moment at which the previous $k-1$, institutions can redeem the corresponding funds from their accounts. This time lag enables us to ensure that if an update to the KYC process needs to be carried out by the most recent FI that onboards the customer (the FI paying $\frac{m}{k}$ to each FI that previously worked with that customer), a proportional part of the update costs is deducted from the original $\frac{m}{k}$ contribution paid by the FI conducting that update. All the FIs working with the customer thus contribute to the cost of this KYC update and pay an amount equal to $\frac{m}{k} - \frac{c}{k}$. The system thus guarantees that all participating FIs that work with a given customer have access to the most recent, updated version of the document package.

This procedure for immediate and automatic proportional cost sharing also allows all FIs to have access to the most recent documents at all times. Assuming that k FIs have onboarded a given customer prior to an update being required, the FI that carries out the KYC update process assigns a value $c$to the update. This value represents the amount, in monetary terms, that the FI has spent on the update. All other $k-1$ FIs are then required to pay $\frac{c}{k}$, which they transfer to the FI that has carried out the update. In this setup, the total cost of onboarding and updating the customer, $m+c$ is proportionally distributed among the k FIs, which pay $\frac{m}{k}+\frac{c}{k}$ each. This process is repeated each time an update is required.

We acknowledge that the implementation of this update-costs element depends on agreement among participating FIs with regard to which documents are being updated. The method applied in the PoC presented in this paper assumes fixed update costs that are dependent on the initial KYC cost m. For each smart contract there are three types of update cost that can be assigned to the update. Each type is a fraction of the initial KYC cost. For purposes of illustration, we choose to use $c = \frac{m}{2}$, $c = \frac{m}{4}$, and $c = \frac{m}{6}$.

After the update has been carried out and the update cost has been distributed proportionally among all onboarding FIs, the total cost of the customer increases from m to $m + c$. Assuming that a total of K FIs have access to the system, all new onboarding FIs are required to pay $\frac{m+c}{j}$, where $j = k + i$ and $i = \{1, 2, ..., K - k\}$.

This system would respect the four (revised) assumptions and the four conditions that we define earlier in this section, and would constitute a significant improvement—in terms of efficiency, applicability, incentive structure, and maintenance costs—over all the previous solutions that propose improving the KYC process by means of DLT.

2.6.3 Technical Description of the Optimized, Dynamic KYC Process

This section shows how we implemented the desired properties of the smart contract as functions programmed in *Solidity,* a programming language for smart contracts deployed on the Ethereum network. Here we describe the essential code for the implementation of the proposed system. The proposed system relies on two smart contracts: the KYC smart contract and the token smart contract. For the token smart contract an open-source code based on ERC-20 recommendations is used (Buterin and Voglsteller, 2015). We made slight adjustments to the ERC-20 code to simplify communications between the two smart contracts.

The smart contract tracks the FIs addresses, the amount of tokens each FI transfers to the contract, and the hash imported initially or updated as a struct Bank.

```
1 struct Bank {
2   address bankAddress;
3   bytes32 bankHash;
4   uint256 balances;
5 }
6 address[] public banks_ids;
```

Figure 10: Smart Contract Structure.

The system records the FIs that have onboarded the customer in a list of addresses as banks_ids. banks_ids is what we have referred to as the list of onboarding institutions in previous sections. When a customer is added to the network the FI that carries out the initial KYC process creates the smart contract for that customer. The PoC uses a separate smart contract to deploy the smart contract used

for KYC purposes. The deployment smart contract uses a function deployContract that takes as input the parameters necessary to deploy the KYC smart contract.

```
1 contract deployCheckHash {
2   function deployContract(address newAddress, uint256 typeOf,
        address _bankAddress, string _hash, address tokenAddress)
        returns (address){
3     return new checkHash(newAddress, typeOf, _bankAddress, _hash,
          tokenAddress);
4   }
5 }
```

Figure 11: Deployment Smart Contract.

The necessary inputs required by deployContract are the address of the customer (newAddress), the cost of the customer ($typeOf$), the address of the FI (_bankAddress), the hash of the KYC documents ($\_hash$), and the address of the token smart contract (tokenAddress). The token contract, as well as all the code, can be consulted in the GitHub Repository "KYC-Optimized-and-Dynamic-System-using-Blockchain-Technology (Tth2549, 2017).

The deployment smart contract uses its inputs to deploy the KYC smart contract as checkHash. The checkHash smart contract uses an initializing function to store the required information and transfer the ownership of the smart contract from the FI to the customer.

The initializing function is only called when the smart contract is created. After ownership has been transferred the function calls storeProof to store the hash in the smart contract initializing function. Next, the FI is added to banks_ids and the hash is linked to the FI to record what hash the first FI imported. Further, the function uses the input typeOf to set the cost of the customer. Note that in this PoC we assume that there only exist three types of customers, which cost 100,000, 200,000, and 300,000 tokens, respectively. These values were chosen arbitrarily for illustration purposes.

The customer is registered as the owner of the contract and therefore reserves the right to erase/kill the smart contract. To do this the customer needs to approach an FI that has access to the network. Ownership of the contract is then transferred from the customer to that FI so the function kill can be called and the contract erased.

```
1  function checkHash(address newOwner, uint256 typeOf, address
       _bankAddress, string _hash, address tokenAddress) {
2    transferOwnership(newOwner);
3    banks[_bankAddress].bankAddress = _bankAddress;
4
5    TokenAddress = tokenAddress;
6
7    bytes32 proof = proofFor(_hash);
8    storeProof(proof);
9    banks[_bankAddress].bankHash = proof;
10   banks_ids.push(_bankAddress);
11
12   if (typeOf == 1) {
13     contract_cost = 100000;
14   }
15   else if (typeOf == 2) {
16     contract_cost = 200000;
17   }
18   else if (typeOf == 3) {
19     contract_cost = 300000;
20   }
21   else throw;
22 }
```

Figure 12: Smart Contract initializing Function.

After the KYC smart contract has been successfully deployed, it can be used by other FIs. The smart contract has a function, payment, that can be called by any FI. The payment function is used to inform FIs of the sum they must pay in order to interact with the smart contract.

```
function transferOwnership(address newOwner) onlyOwner {
  owner = newOwner;
}

function kill() {
  if (msg.sender == owner) {
  selfdestruct(owner);
  }
}
```

Figure 13: Smart Contract Function transferOwnership.

The payment function simply takes the cost of the customer and divides it by the number of FIs that have previously onboarded that customer plus one. If the FI accepts the amount to be paid, it can proceed to pay the given amount and use the smart contract further. The smart contract uses two functions to transfer the tokens—tokenFallback and payContract. The function tokenFallback stores the tokens in the smart contract and then uses payContract to distribute these tokens accordingly to other FIs.

```
function payment() constant returns (uint256){
  return contract_cost/(banks_ids.length+1);
}
```

Figure 14: Smart Contract Function transferOwnership.

Note that only after the tokenFallback function has verified that the amount transferred to the smart contract is correct does it call payContract.

```
function tokenFallback(address from, uint256 amount, bytes data){
  if (amount == contract_cost/(banks_ids.length+1)){
    banks[from].balances += amount;
    banks[from].bankAddress = from;
    payContract();
    banks_ids.push(from);
  }
  else throw;
}
```

Figure 15: Smart Contract Function tokenFallback.

```
function payContract() {
  TokenERC20 t = TokenERC20(TokenAddress);
  uint256 totalCost = contract_cost/(banks_ids.length+1);
  for (uint i = 0; i < banks_ids.length; i++){
    t.transfer(banks_ids[i],totalCost/banks_ids.length);
  }
}
```

Figure 16: Smart Contract Function payContract.

The function payContract uses a simple loop to transfer the payment to all previous FIs. Note that the function tokenFallback adds the new FI to the list of onboarding institutions, banks_ids, after the payContract function has been called. After the FI has paid the contract and the payment has been distributed, the FI can use the function checkDocument to compare the hash stored in the smart contract to the hash it has generated from the documents supplied by the customer.

```
function checkDocument(string document) alreadyPaid constant
    returns (string) {
  bytes32 proof = proofFor(document);
  return hasProof(proof);
}
```

Figure 17: Smart Contract Function checkDocument.

The function checkDocument uses the modifier alreadyPaid to ensure that the FI calling it is on the list of onboarding institutions—that is to say, that the FI has already paid its share.

```
modifier alreadyPaid {
  if (banks[msg.sender].bankAddress != msg.sender) throw;
}
```

Figure 18: Smart Contract Modifier alreadyPaid.

If the FI has paid its share, checkDocument converts the input hash to the same format as the hash stored in the smart contract, using proofFor.

```
function proofFor(string document) constant returns (bytes32) {
  return sha256(document);
}
```

Figure 19: Smart Contract Function proofFor.

Further, the function hasProof is used by the smart contract to compare the hash that results from hashing this document with the hash stored in the smart contract.

```
function hasProof(bytes32 proof) constant returns (string) {
  if (proofs.length == 0) return "No data here.";
  if (proofs[proofs.length-1] == proof){
    return "Data is correct!";
  }
  else {
    for (uint256 i = 0; i < proofs.length; i++) {
      if (proofs[i] == proof) {
        return "Data is old, has been approved before." ;
      }
    }
  }
  return "Data has never been approved!";
}
```

Figure 20: Smart Contract Function hasProof.

The hasProof function has three possible returns: “Data is correct!” if the hash is a match, “Data is old, has been approved before” if the hash that the FI is comparing has been updated by another FI and the new FI is using old documents, and “Data has never been approved!” if the hash does not match and has never been used before. If the documents used for the KYC process need to be updated, the FI can update the hash in the smart contract using the function notarize.

```
function notarize(string document) alreadyPaid {
  bytes32 proof = proofFor(document);
  storeProof(proof);
}
```

Figure 21: Smart Contract Function notarize.

Notarize converts the hash of the updated document to SHA256 and then adds the new, resulting hash to storage in the smart contract using storeProof. The function notarize notes which FI updated the hash so there is a record of which FI has imported each specific hash. Additionally, the function keeps track of all the hashes used for the customer, in storeProof.

```
1 function storeProof(bytes32 proof) internal {
2   proofs.push(proof);
3   if (banks_ids.length > 1){
4     banks[msg.sender].bankHash = proof;
5   }
6 }
```

Figure 22: Smart Contract Function storeProof.

2.7 Conclusion

In this paper we propose a refined, dynamic, blockchain-technology-based KYC system that reduces the costs of the KYC process, allows these costs to be shared proportionally by participating FIs, eliminates the need for a TTP to manage permissions in the system, and conducts dynamic updates with regard to the status of FIs' customers over time. Further, we develop a PoC that can be used by FIs and regulators to implement the proposed system and to explore variations of the system. This system is based on previous system proposals and emerged through the application of DSR. Specifically, the system development process employed a series of loops of design, evaluation, and demonstration that served to improve the system's applicability to a real-life corporate environment and to resolve the inefficiencies of previously proposed systems.

The major contribution of the system we propose has been that it eliminates the need for a TTP, making the system truly decentralized, and that it makes possible a distributed data storage architecture that is independent of the blockchain architecture, which makes implementation more cost efficient and easier for FIs. In our system the blockchain is only used to grant and manage the distributed database's reading permissions. This strengthens the incentive structure for participating FIs, ensuring that they act in the way in which they are meant to act because of an intrinsic impulse and not simply due to the fear of being supervised by the regulator. Our system has made another vital contribution in that it allows each participating FI to dynamically update each customer's status, such that if an FI identifies—for example—a flaw with regard to the legality of a customer's activities, it can revise that customer's

status and propagate this information through the system to those other FIs that work with that customer. The implications of this are, in fact, crucial because this feature not only allows FIs to revise the status of any given customer, it also increases the quality of the information—in the form of KYC documentation—available to the network, which ensures that all participating FIs remain up-to-date in terms of the validity of the KYC status of any customer.

Additionally, we prove the concept by means of an artifact—coded in the language *Solidity*—that can be easily used by any interested individual to test and develop the concept, implement it in an experimental environment, and further develop it and adapt it in order improve its applicability and usefulness. We are convinced that the conceptual system and the PoC that we propose here can serve to improve the existing KYC process and that they constitute one necessary further step toward the adoption of blockchain-based systems in the corporate environment.

Essay 3

An Urn Filled with Bitcoins:

New Perspectives on Proof-of-Work Mining

An Urn Filled with Bitcoins: New Perspectives on Proof-of-Work Mining

José Parra-Moyano
University of Zurich
Switzerland
jose.parramoyano@uzh.ch

Gregor Reich
University of Zurich
Switzerland
gregor.reich@uzh.ch

Karl Schmedders
University of Zurich
Switzerland
karl.schmedders@uzh.ch

May 2019

Abstract

The probability of a miner finding a valid block in the bitcoin blockchain is assumed to follow the Poisson distribution. However, simple, descriptive, statistical analysis reveals that blocks requiring a lot of time to find–long blocks–are won only by miners with a relatively higher hash power per second. This suggests that relatively bigger miners might have an advantage with regard to winning long blocks, which can be understood as a sort of "within block learning". Modelling the bitcoin mining problem as a race, and by means of a multinomial logit model, we can reject that the time spent mining a particular block does not affect the probability of a miner finding a valid version of this block in a manner that is proportional to her size. Further, we postulate that the probability of a miner finding a valid block is governed by the negative hypergeometric distribution. This would explain the descriptive statistics that emerge from the data and be aligned with the technical aspects of bitcoin mining. We draw an analogy between bitcoin mining and the classical "urn problem" in statistics to sustain our theory. This result can have important consequences for the miners of proof-of-work cryptocurrencies in general, and for the bitcoin mining community in particular.

Keywords: Blockchain, Bitcoin, Discrete Choice Modeling

Note: A version of this paper is available as Parra Moyano, J. Reich, G., and Schmedders, K., Urns Filled with Bitcoins: New Perspectives on Proof-of-Work Mining, SSRN, 2019, doi: https://dx.doi.org/10.2139/ssrn.3399742.

3.1 Introduction

Bitcoin is the electronic peer-to-peer currency, payment, and economic system first proposed by Nakamoto (2008) that has since its introduction attracted the attention of scholars and practitioners. The paper by Nakamoto (2008) opened a new stream of research in the literature and gave rise to both the industry of "blockchain"—the underlying technology behind bitcoin—and "cryptocurrencies". The market capitalization of bitcoin and all existing cryptocurrencies reached 800 billion US dollars in 2018 (CoinMarketCap, 2019) and the size of the blockchain technology market is predicted to reach an annual value of more than 23 billion US dollars by 2023 (Grand View Research, 2018).

One of the properties that made bitcoin unique at the time of its introduction was the fact that it uses a system called "mining", which is conducted by machines that implement the bitcoin protocol and are called "miners" in order to reach a secure, tamper-resistant consensus with regard to the transactions made in the system, as well as to generate and introduce new bitcoins—units of the currency—into the system (Antonopoulos, 2014). Miners conduct a trial-and-error process to find a "valid" block—a piece of information that is linked to the past history of all the bitcoin transactions and that, among other things, generates a predefined number of new bitcoins. The probability of a miner finding a valid block is determined by her individual "hash rate per second" (the number of trials that she can conduct per second) and the mining difficulty, D, which is a predefined number that is computed to keep the average time required until any of the participating miners finds a valid block at 10 minutes. Each block contains a reference to all the previous blocks that have been found. This forces a chronological order on the mining process and results in the fact that miners compete simultaneously to find a valid version of the "next block". The mining process resembles a race, in which miners compete to become the first to find the next valid block. As soon as a miner finds the next valid block, the winning miner broadcasts the valid block to all the other participating miners such that a new race containing this new valid block can start. Miners can collaborate with each other by gathering in "mining pools". For the sake of our analysis we use "miner" whenever we refer to either a miner or a mining pool and "mining pool" when we refer specifically to a mining pool.

It is broadly accepted that the probability of finding a valid block—the "arrival rate" of valid blocks—is a Poisson process (Bowden, Keeler, Krzesinski, & Taylor, 2018). This is assumed by authors including Nakamoto (2008), Rosenfeld (2011), Eyal and Sirer (2013), Decker and Wattenhofer (2013), Rosenfeld (2014), A. K. Miller and LaViola (2014), Sapirshtein, Sompolinsky, and Zohar (2015), Göbel, Keeler, Krzesinski, and Taylor (2015), Lewenberg, Bachrach, Sompolinsky, Zohar, and Rosenschein (2015), Houy (2016), Cocco and Marchesi (2016), Solat and Potop-Butucaru (2016), Beccuti and Jaag (2017), Chiu and

Koeppl (2017), Dimitri (2017), Aggarwal, Brennen, Lee, Santha, and Tomamichel (2018), L. Cong, Li, and Wang (2018), Hayes (2019), Easley, O'Hara, and Basu (2019), L. W. Cong, He, and Li (2019), and Wang et al. (2019). The assumption that the mining process follows the Poisson distribution implies that the miners' probabilities of winning remain constant throughout the race (i.e., throughout the time that elapses between the moment at which a miner starts trying to find the next valid block and the moment at which any miner finds and broadcasts that valid block). This also means that two miners with identical and constant hash rates per second have identical and constant probabilities of winning throughout the race, regardless of the moment at which they start mining and regardless of the number of failed trials they have previously conducted for this particular block. Or, phrased differently, the assumption implies that the length of the race (the time that passes between the moment at which a miner starts mining a particular block and that at which a valid version of this block is found) does not influence the probability of the miner winning. This assumption is used by the Bitcoin protocol to set the mining difficulty, D, which determines the expected time required until a valid block is found. This assumption is also used by all the miners that can participate in the mining process in order to determine their expected returns on mining and hence to decide whether to participate in the mining process or not (entry–exit decision). Further, this assumption is used by cryptocurrency exchanges to estimate the hash rate of mining pools, and by cryptographers to determine the security level of all bitcoin-like (proof-of-work) blockchains. We theorize and suggest, however, that the probability of winning a block increases during the time that a miner spends mining this block. This implies that the probability of winning a block does not follow the Poisson distribution and we suggest that it could follow the negative hypergeometric distribution instead. We test our postulate by means of a multinomial logit model similar to the one used by Bolton and Chapman (1986) to model horse races and conclude that we cannot reject the hypothesis that the probability of winning a block increases along the time that a miner spends mining this block. This result implies that the probability of winning varies throughout the time that miners are mining the same block, that bigger miners have a higher probability of winning longer blocks (or races), that it might be in the interest of smaller miners to stop mining when they reach a certain time threshold, that cryptocurrency exchanges need to change the way in which they calculate mining pools' hash rates and "luck"[1], and that the way in which the Bitcoin difficulty, D, is determined needs to be adapted. Further, it implies that the expected return on mining for relatively small miners is smaller than is assumed, whereas the expected return on mining for relatively large miners is higher than is assumed. That the probability of winning a block follows the hypergeometric distribution is a plausible explanation to this phenomenon.

In Section 3.2 we describe the fundamentals of the mining processes of bitcoin as well as the observations

[1]Pool luck is a parameter that emerges from the difference between the expected success rate of mining (given the hash power) and the observed success rate of mining.

and assumptions made by other scholars with regard to the mining process following the Poisson distribution. Further, we propose and explain why we consider that the mining processes follows the negative hypergeometric distribution, making an analogy between bitcoin mining and the classical "urn problem" in statistics. In Section 3.3 we describe the data that we use and present the descriptive statistics that motivate our postulate. In Section 3.4 we present the multinomial logit model that we use to model the bitcoin mining process. In Section 3.5 we show the results of the estimation. In Section 3.6 we discuss the implications of our results and we conclude.

3.2 Bitcoin Mining

In this section we briefly describe the concept of hashing, we present the fundamentals of bitcoin mining, we describe the assumptions scholars make in order to conclude that the arrival rate of valid blocks follows the Poisson distribution, and we describe why we consider that the probability of winning a block could in fact follow the negative hypergeometric distribution.

3.2.1 Fundamentals of the SHA-256 Function

The SHA-256 (Secure Hash Algorithm 256) function is a cryptographic, one-way compression function. The domain of the SHA-256 function is composed by any string of length up to 2^{64} bits. This implies that the domain of the SHA-256 function contains $2^{(2^{64}-1)}$ possible, different inputs. This is the set of possible inputs to the function. The range (support) of the SHA-256 function encompasses 2^{256} possible 256-bit strings. All the strings that result from the SHA-256 function have a length of 256 bits. The result of a hashing function is called a "hash". The SHA-256 function is deterministic, such that the same input always yields the same output. Altering one bit in the input passed through a hashing function completely alters the resulting output. The SHA-256 function is a one-way function, which means that the original data can not be retrieved from the resulting data (i.e., in order to find the input that yields a particular output, only a trial-and-error process can be conducted). Since the SHA-256 function cannot be inverted, it is impossible to anticipate which input is going to yield a particular output. Courtois, Grajek, and Naik (2014a) describe further details of the SHA-256 function applied to the context of bitcoin mining.

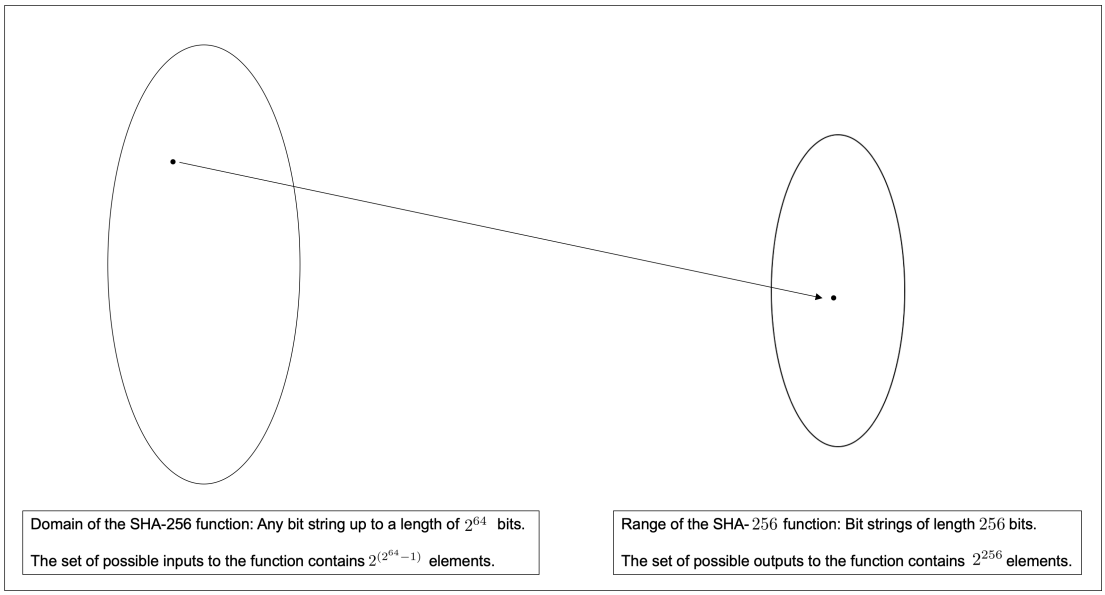

Figure 24: Illustration of the domain and range of the SHA-256 function.

3.2.2 Fundamentals of Bitcoin Mining

The underlying technology behind bitcoin is called the blockchain. A blockchain is a peer-to-peer ledger that stores information in packages called blocks. Every block is linked to each other block by containing the hash that results from hashing the previous block. Blocks have a size limit set to 1 MB [2]. As explained by Courtois, Grajek, and Naik (2014b), blocks contain the following information:

- **Version number**: An integer representing the version number of the bitcoin software. This number defines the rules governing the blocks.
- **The hash of the previous block**: All the input of the previous block is hashed, such that the resulting hash can be included in the current block. This links each block with the previous one.
- **The Merkle root**: A Merkle root is part of a Merkle tree and makes a reference to the transactions that are stored in this block. The Merkle root can be understood as an aggregated hash of all the transactions contained in the block. It is important to note that one of the transactions stored in the block is written by the miner writing the block itself. This transaction is the transaction that creates and assigns the new bitcoins to the miner itself writing the block. This is the way in which miners are compensated for their mining costs.
- **Timestamp**: The time at the moment at which the block was written.

[2]Due to the Segregated Witness (SegWit) protocol upgrade implemented in 2017, blocks that are larger than 1 MB are accepted if they fulfill a set of requirements. For more information on SegWit see https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki.

- **Target**: A global variable target, also known as Difficulty, D, that determines the blocks that will be considered "valid".
- **Padding + Len**: Two constants required by the SHA-256 hash function.
- **Nonce**: The nonce is a 32-bit number chosen by the miner.

The objective of miners is to take the inputs given by the network (the version number, the hash of the previous block, the timestamp, the target, and the Padding + Len), incorporate the Merkle root with transactions waiting to be included in the blockchain, hash it using the SHA-256 function, and hash the resulting 256-bit string again using the same SHA-256 function, such that the second resulting hash starts with a number of zeroes larger than that determined by the target (or Difficulty)[3]. As soon as a miner finds a nonce that, together with the input given by the network and the information that she has written, is hashed twice and yields a 256-bit string starting with the minimum accepted amount of zeroes, she has found a valid block. At the moment at which a miner finds a valid block, it is in her interest to broadcast this valid block to the network such that the other miners can verify that the input and the nonce provided by the winning miner, once hashed, in fact start with the desired number of zeroes and accept the broadcasted block as valid. Once the block is accepted as valid, a new problem using the hash of this newly accepted block as the one of the inputs for the next block starts for all the miners. Since the SHA-256 function cannot be inverted, it is impossible to anticipate which input (which nonce) is going to yield a particular output. This is the cornerstone of proof-of-work mining: miners have to take the input given by the previous block, add the corresponding Merkle root as well as the further required information, and try many different 32-bit random nonces such that once that they are hashed together (the nonce and the rest of the information written in the block) twice, the result of the function yields a number starting with at least a certain amount of zeroes. Should the nonces be exhausted for a potentially valid block, the miner just includes a "superNonce" in the coinbase of the Merkle tree—a field that can be written, with no format specification, by the miner—generating a new version of the potentially valid block. Then, the miner repeats the process and can try a new set of nonces for this new potentially valid block. As explained by Courtois et al. (2014a), since the size of this extraNonce is only limited by the size of block itself, it can be as large as required as long as the block size is within protocol limits. Adding the extraNonce alters the resulting hash of the block in which the miner is working without interfering with its correctness, and therefore allows the miner to try a new set of 32-bit nonces to find a valid block.

The process of mining (this trial-and-error process of testing many different nonces such that the result yields a number below a predefined threshold) is carried out by specialized hardware that consumes elec-

[3]There are methods to speed up this double-hashing procedure. See Hanke (2016) and Courtois et al. (2014b).

tricity. The power of mining hardware is measured in hashes per second (i.e., the number of hashing operations that the machine can achieve per second). Miners are motivated to mine due to the "block reward" and "block fees". The block reward is the amount of newly created bitcoins, defined by the protocol, that the miner can create and assign to herself in a transaction written in each potentially valid block. This transaction becomes effective and accepted by the network only if the potentially valid block becomes effectively valid. Note that the transaction that creates these new bitcoins and assigns them to the miner is one of the transactions that the miner writes in the block, such that this transaction is part of the Merkle tree contained in each block[4]. Further, the transactions that are made by bitcoin wallets and that are sent to the so-called mempool for the miners to pick them and include them in their blocks can contain a fee that goes to the miner who includes them in a valid block. Hence, once a miner finds a valid block, she receives the newly created bitcoins and the fees of the transactions that she includes in her block.

Figure 25 illustrates the creation of a block. In Figure 25 three miners are competing to find the next valid block—namely, Miner A, Miner B, and Miner C. The last valid block found in this example is Block 8. Each of the three miners uses the hash of Block 8, together with the target, the version, and the Padding + Len, in order to write a potentially valid block. A potentially valid block is a block that fulfills all the requirements to be a valid block with the exception of the nonce. The potentially valid blocks written by each miner are different from each other because despite the fact that the target, the hash of the previous block, and the version and the Padding + Len are the same for all, each miner writes a different timestamp, and includes a different set of transactions (different Merkle root) on it. Once each miner has written a potentially valid block, she tries different nonces until the hash of the hash of a version of the potentially valid block starts with the required number of zeroes. Once this occurs, the hash of this block (Block 9 in our example) becomes, together with the target, the version, and the Padding + Len, the common basis for all the miners to write (each of them) their next potentially valid block (Block 10 in our example).

Mining is a random process. Miners take the last valid block, use it to write a new block that contains the required information, include a nonce in the block, and hash the block twice, hoping to get a valid hash (with at least as many zeroes as required by the target). If miners don't succeed in this process, they increment the nonce by one, add it again to the block, and hash that new block twice trying to get a valid hash at this new attempt. This process is repeated continuously until a hash less than the target value is found by any participating miner. Miners therefore follow a stochastic process to find valid blocks, in which they write a block and then start hashing the block with different nonces or superNonces until a

[4]The amount of the reward started at 50 bitcoins and is halved every 210,000 blocks. This occurs approximately every four years.

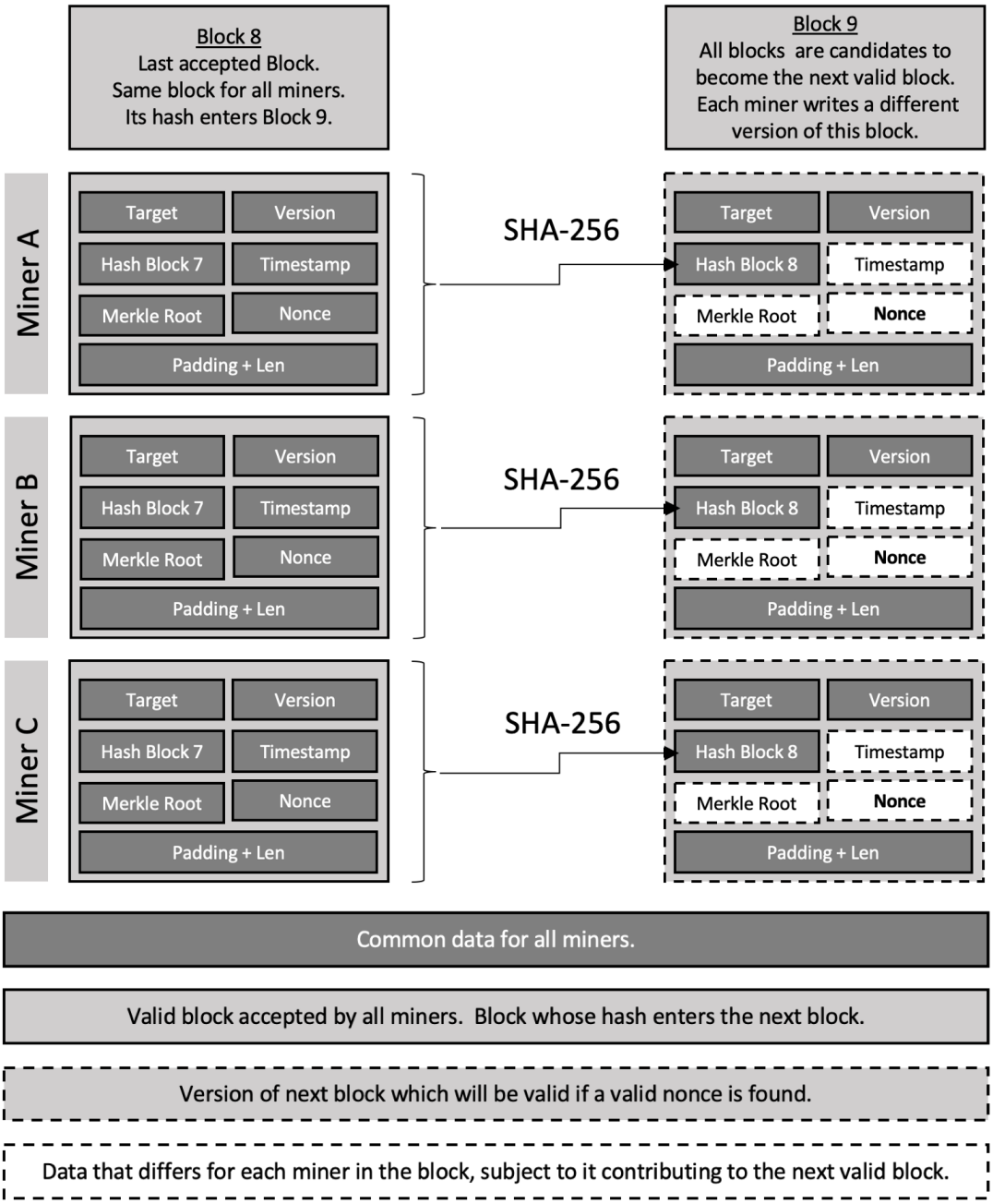

Figure 25: Illustration of the blockchain and the different versions of a block that can become the next valid block.

valid block is found. Miners don't try the same nonce with the same block twice since they already know that it results in an invalid hash.

3.2.3 Bitcoin Mining as a Poisson Process

Nakamoto (2008) introduces bitcoin and the process of creating new units of the cryptocurrency by hashing the previous block and comparing the result to a certain threshold. He suggests—though he does not explicitly state—that valid blocks are found according to the Poisson distribution. Probability events that follow the Poisson distribution can occur n times in an interval. The average number of events in an interval is designated by λ, which is also called the rate parameter. In the Poisson distribution the probability of observing k events in an interval is given by the equation

$$P(k) = e^{-\lambda}\frac{\lambda^k}{k!}.$$

As described by Rosenfeld (2011), the bitcoin protocol sets the target value (the minimum number of zeroes with which a hash will be considered valid) such that every hash has a probability $\frac{2^{16}-1}{D2^{48}}$ of yielding a valid block. For the sake of simplicity, we—like other scholars—approximate this value by $\frac{1}{D2^{32}}$. This implies that the Poisson parameter that determines the probability of winning of miner i over time t expressed in seconds (the probability of finding a valid block after mining for time t) can be written as

$$\lambda_i = \frac{h_i t}{D2^{32}},$$

$h_i t$ being the hash rate of miner i and D the Difficulty parameter. Therefore, a miner i mining at a rate of h_i hashes per second for time t (time expressed in seconds) has an expected rate of winning $\frac{h_i t}{D2^{32}}$. This assumption is used by many scholars to study different aspects of the bitcoin network. Eyal and Sirer (2013) study the incentive structures for selfish mining in the Bitcoin network (a process tried by some miners to leverage the time advantage they have when they find a valid block) and directly assume the Poisson distribution for the arrival rate of blocks. Decker and Wattenhofer (2013) study the propagation of transactions and blocks through the bitcoin network in order to understand the creation of forks. In their analysis they specifically state that the proof-of-work mining process is a Poisson process, in which the time difference between blocks follows an exponential distribution. Rosenfeld (2014) states that while the qualitative nature of bitcoin mining is well understood, there is widespread confusion about its quantitative aspects and how they relate to attack vectors and their countermeasures. He looks at the stochastic processes underlying typical attacks and their resulting probabilities of success, assuming again

that the number of blocks found by miners follows the Poisson distribution. A. K. Miller and LaViola (2014) present a formal model of anonymous, synchronous processes that communicate using one-way public broadcast, showing that the Bitcoin protocol achieves consensus using this model. For their proofs, they assume that the total number of puzzle solutions found by the network (i.e., the total number of valid blocks found by a miner) follows the Poisson distribution. Göbel et al. (2015) study the effect of propagation delay on the evolution of the Bitcoin network, and use a spatial Poisson process model to study the values computed by Eyal and Sirer (2013). Lewenberg et al. (2015) describe that if blocks in a blockchain (bitcoin or others) are created at a high rate compared to their propagation time in the network, many conflicting blocks are created, which can negatively affect the transaction throughput of the system; they also propose an alternative structure to the chain that allows for operation at higher rates. For both their analysis and their solution, they assume the Poisson distribution for the arrival rate of blocks. Houy (2016) deals with the mining incentives in the Bitcoin protocol, which he defines as a speed game between the miners, which differ from each other in terms of the computational power (hash rate per second) with which they try to find a valid block. In his definition of their game, as well as in the analytic search for Nash equilibria, he assumes that the process of finding a valid block can be modelled as a random variable following the Poisson distribution. Cocco and Marchesi (2016) present an agent-based artificial market model of the Bitcoin mining process, which they use to model the economy of the mining process. They define the probability of mining a valid block as the relative hashing power of the miner at every point in time $r_i(t)$ divided by the total hashing power of all the miners in the network at the same time $r_{Tot}(t)$, such that the number of bitcoins, b, that a miner can expect to mine is the probability of winning multiplied with the total reward, B, of newly created bitcoins: $b = \frac{r_i(t)}{r_{Tot}(t)} B$. Note that the probability of winning remains constant over time such that time plays no role in the expected number of bitcoins won. Solat and Potop-Butucaru (2016) propose and theoretically analyze a solution for the bitcoin selfish mining attack. They propose a solution to prevent such attacks by exploiting the Poisson nature of the proof-of-work mining protocol. Beccuti and Jaag (2017) model the bitcoin mining process as a game of imperfect information, in which miners have to choose whether or not to report their success (selfish mining). They show that the game has a multiplicity of equilibria and that the minimum requirement to find it optimal not to report a block decreases with the number of "selfish" miners. For their analysis, they assume that success in bitcoin mining follows the Poisson distribution with the parameter proposed by Rosenfeld (2011). Chiu and Koeppl (2017) study the optimal design of cryptocurrencies and assess quantitatively how well such currencies can support bilateral trade. They propose a design for cryptocurrencies that reduces mining and relies exclusively on money growth rather than transaction fees to finance mining rewards. For the analysis of the optimality of their design they assume that the probability of a miner finding a valid block is proportional to the fraction of computational

power that that miner owns, utilizing the results of Rosenfeld (2011). Dimitri (2017) presents a game-theoretic framework, assuming complete information in order to model Bitcoin mining. Among other findings, he finds that in the unique pure strategy Nash equilibrium of the game the optimal amount of energy consumption of miners depends on the reward for solving the puzzle (finding a valid block). For the definition and the analysis of the game he assumes the same probability of winning as Rosenfeld (2011). Aggarwal et al. (2018) investigate the exposure of Bitcoin, and other cryptocurrencies, to attacks by quantum computers and suggest that the proof-of-work mining protocol used by Bitcoin is relatively resistant to the threat posed by quantum computers' substantial speedup, mainly because specialized ASIC miners are extremely fast compared to the estimated clock speed of near-term quantum machines. For the success rate of mining they take a probability of the same form as Rosenfeld (2011). L. Cong et al. (2018) provide a dynamic asset-pricing model of cryptocurrencies. For their model, they define the law of motion of token supply (the result of the mining process) as an exogenous stochastic Markov process. They also present an alternative formulation in which the token supply follows the Poisson distribution, as "seen in Bitcoin's supply schedule". They state that formulating the process as a Poisson process has the advantage that equilibria between two Poisson arrivals still have only one state variable, which allows the authors to solve the model by backward induction, starting from the asymptotic future where token supply has plateaued and moving back sequentially in the Poisson time. Hayes (2019) proposes and tests a cost-of-production model for valuing bitcoin. In order to calculate the expected number of bitcoins that a miner can produce (which is crucial for the miner to decide whether she participates in the mining process or not), he implicitly assumes the Poisson distribution for the arrival rate of blocks. Easley et al. (2019) investigate the role that transaction fees play in the Bitcoin blockchain's evolution from a mining-based structure to a market-based economy. They do so by developing a game-theoretic model to explain the factors leading to the emergence of transactions fees, as well as to explain the strategic behavior of miners and users. Their model also highlights the roles played by mining rewards and by volume, and examines how microstructure features such as exogenous structural constraints influence the dynamics and stability of the Bitcoin blockchain. In order to do so, they assume that, independently, for each miner working on the problem, valid blocks arrive according to the Poisson distribution. Wang et al. (2019) provide a systematic vision of the organization of blockchain networks. By emphasizing the unique characteristics of incentivized consensus in blockchain networks, their review of the existing consensus protocols is focused on both the perspective of distributed consensus system design and the perspective of incentive mechanism design. They also assume the arrival rate of blocks to follow the Poisson distribution. L. W. Cong et al. (2019) study how the rise of centralized mining pools for risk sharing affect the relationships between competing miners and the energy consumption of proof-of-work-based blockchains. They state that in proof-of-work mining the probability of finding a solution is not affected by the number of trials previously attempted.

This is what they call the "well-known 'memoryless' property" of proof-of-work mining, which "implies that the event of finding a solution is captured by a Poisson process with the arrival rate proportional to a miner's share of hash rates globally" as described by Eyal and Sirer (2013) and Sapirshtein et al. (2015). By assuming that the arrival rate of blocks follows the Poisson distribution, all these authors are assuming (as nicely stated by L. W. Cong et al. (2019)) that mining is a "memoryless" process, and therefore that the probability of finding a valid block is independent of the previously made attempts.

Grunspan and Perez-Marco (2017) correct the analysis given in Nakamoto (2008) regarding the success of double-spend attacks on the bitcoin blockchain, and give a closed-form formula for the probability of success of a double-spend attack. In doing so, they assume that the number of blocks $N(t)$ mined at time t follows the Poisson distribution. Grunspan and Perez-Marco (2017) do not revise or study the arrival rate of blocks, but demonstrate that one of the characteristics of the bitcoin blockchain that emerges from this arrival rate—namely, the success of double-spend attacks—differs from what was previously assumed in the literature. Their work is especially important for us, since while it still accepts that the arrival rate of blocks follows the Poisson distribution, it challenges the assumption implied by Nakamoto (2008) that the success of a double-spend attack also follows the Poisson distribution.

Bowden et al. (2018) challenge the assumption of the block arrival rate following the Poisson distribution, and based on a stochastic analysis of the block arrival process demonstrate that this is not the case. They present a refined mathematical model for block arrivals, focusing on both the block arrivals during a period of constant difficulty and how the difficulty level evolves over time. Their work leaves the question of "how does the arrival rate of blocks really behave?" open.

3.2.4 Bitcoin Mining as a Negative Hypergeometric Process

The negative hypergeometric distribution arises in schemes of sampling without replacement. If in the total population of size N there are M elements that are considered a "success" and $N - M$ elements that are consider a "failure", and we sample elements out of the population without replacement until the number of "successes" reaches a fixed number m, then the random variable X—the total number of "failures" in the sample—follows the negative hypergeometric distribution $X \sim \text{NHG}\ (N,\ M,\ m)$. The probability mass function (PMF) of a random variable X that follows the negative hypergeometric distribution is given by

$$\Pr(X = k) = \frac{\binom{k+m-1}{k}\binom{N-m-k}{M-m}}{\binom{N}{M}}.$$

Using the negative hypergeometric distribution, and X being a random variable counting the samples number at which the *k-th* success is obtained when sampling without replacement from a set of N objects of which M are considered a success, X follows a negative hypergeometric distribution. In that case, the probability mass function is given by

$$\Pr(X = x) = \frac{\binom{M}{k-1}\binom{N-M}{x-k}}{\binom{N}{x-1}} \times \Big(\frac{M-k+1}{N-x+1}\Big).$$

For more details on the negative hypergeometric distribution see Johnson and Kotz (1969) and G. K. Miller and Fridell (2007).

As we have described, a miner takes an input related to the past history of all transactions as given. Further, she writes a block containing this input, as well as the additionally required data (Merkle tree, timestamp, etc.). Once this information is written in the block, the miner includes a nonce in it and uses all this data together as the input of the SHA-256 function. As a result of this hashing, the miner gets one of the 2^{256} possible 256-bit strings that can result from the SHA-256 function. She then takes the hash resulting from this operation and hashes it again in order to get another of the 2^{256} 256-bit strings, hoping that this second hash starts with the required number of zeroes. Whenever the second hash does not fulfill the requirements of a valid block, the miner changes the nonce in the block and repeats the double-hashing process. Since trying twice a nonce that results in no valid hash would represent a waste of resources, miners do not repeat a nonce with the same block. Therefore, given the relevant information written in a block, a miner tries successive nonces until she or another miner finds a valid block. Should the nonces be exhausted, the miner just includes the previously described "superNonce" in the block and repeats the process. This process is illustrated in Figure 26.

Both the domain and the range of the SHA-256 function are finite. Since miners conduct a double-hashing process to find a valid block, and don't repeat a nonce that is known to fail for a given block, we postulate that the probability of finding a valid block increases with the number of previously tried nonces and, therefore, the arrival rate of valid blocks follows the negative hypergeometric distribution, such that the probability of a miner finding a valid block is not only be dependent on her hash rate but also on the number of previously non-valid nonces found for this particular block. In other words, we postulate that while proof-of-work mining is still a "memoryless" process *between* blocks, this does not hold *within* blocks.

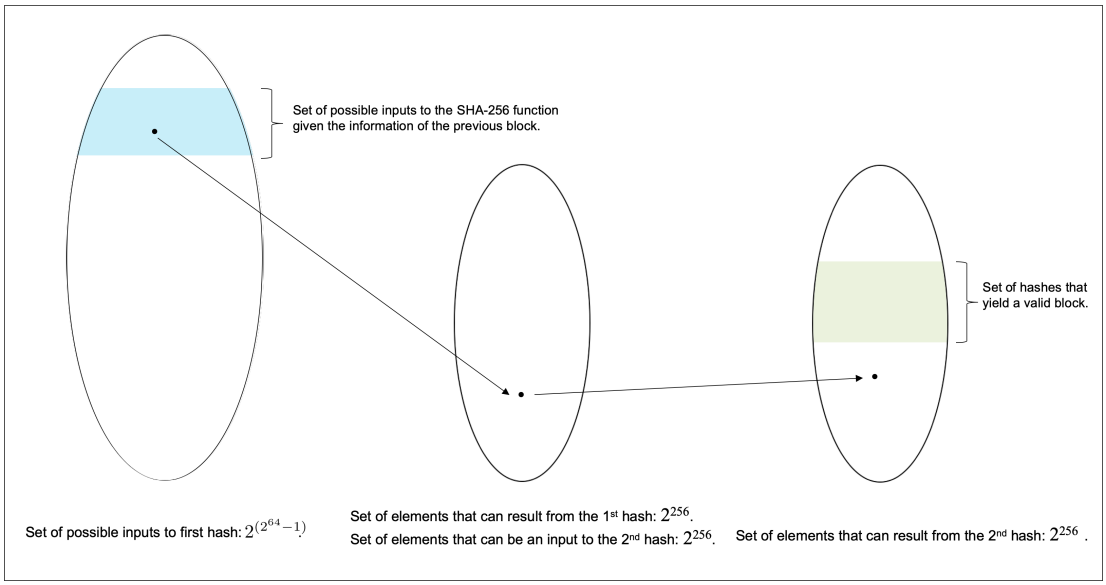

Figure 26: Illustration of the finite domains and ranges of the double-hashing process conducted by bitcoin miners using the SHA-256 function.

Proposition 1 *The number of nonces that needs to be tried until the first valid block is found is a random variable X that follows the negative hypergeometric distribution $X \sim NHG\ (N,\ M,\ 1)$, with N being the number of potentially valid blocks and M the number of potentially valid blocks that, hashed twice, result in a hash below the target.*

In the context of Bitcoin mining, the number of objects with the winning feature equal to M is the number of hashes that start with the required number of zeroes out of all the possible inputs to the SHA-256 function, N, that are contained in the domain of the SHA-256 function[5]. Both N and M are fixed. N equals 2^{256} and M is fixed for each block since the Difficulty, D, that determines it is contained in the header of the block and given by the protocol for each block. The number of draws m made by a miner is the number of nonces tried for a particular block that result in a non-valid hash. No nonce that has previously resulted in a non-valid hash is repeated. The number of observed successes m is set to one since nonces are tried one after the other and as soon as a valid one is found the problem is solved and miners stop trying to find a valid nonce for their block.

[5]Given the fact that miners conduct double-hashing, we could argue that N is the range of the SHA-256 function and not its domain. This does not alter our result, due to the fact that the SHA-256 function is deterministic. Further, since the domain of the SHA-256 function is significantly bigger than its range, one could postulate that collisions will occur. In the context of the SHA-256 function a collision would occur whenever two different inputs to the SHA-256 function (two elements of its huge domain containing 2^{64^2-1} elements) yield the same hash (one of the elements in the huge but smaller range of the function, which contains 2^{256} elements). While this is theoretically possible given the fact that the SHA-256 function is non-injective, collisions have not yet been observed. The SHA-256 function is assumed to be surjective but this has not yet been proven.

Proposition 2 *The probability mass function of finding a valid block on the xth attempt is given by*

$$\Pr(X = x) = \frac{\binom{N-M}{x-1}}{\binom{N}{x-1}} \times \Big(\frac{M}{N-x+1}\Big),$$

with N being the number of potentially valid blocks and M the number of potentially valid blocks that, hashed twice, result in a hash below the target.

Should this be the case, the probability of finding a valid block would increase with the number of failed attempts for this block, such that the longer it takes to find a block (i.e., the longer the time that has elapsed since the broadcasting of the last block), the higher the probability of success of bigger pools (pools that can conduct more hashes per second) compared to that of smaller pools. This would occur since bigger miners could increase their probability of winning faster than smaller miners, following the PMF described in Proposition 2.

3.2.5 Resemblance between Bitcoin Mining and the Urn Problem

In probability and statistics, an urn problem is a thought experiment in which an event (e.g., success) and its complement (e.g., no success) are represented, respectively, by balls of two different colors (e.g., white balls for success and black balls for no success) contained in an urn. In this thought experiment, an individual draws balls from the urn until she draws a ball of the color she is looking for (white). In the easiest version of the experiment, the individual has two options when drawing a ball of the color that does not represent the event that she is looking for: either return the (black) ball to the urn, or not return it to the urn. In the first case, the probability of drawing a ball representing the event (a white ball in our example) remains constant throughout the many consecutive unsuccessful events due to the fact that the individual always returns a ball representing a lack of success (the black ball in our example) to the urn, leaving the probability of success constant across successive attempts. However, in the second case, the probability of drawing a ball representing the event (a white ball in our example) does not remain constant across the many consecutive unsuccessful events due to the fact that the individual does not return a ball representing a lack of success (the black ball in our example) to the urn, therefore altering the proportion of successful events in the urn after each draw. The first case is called "drawing with replacement" whereas the second case is called "drawing without replacement". In the case of "drawing without replacement", in which balls representing unsuccessful events are not returned to the urn, the probability of drawing a ball representing a success after a fixed number of failures follows the negative hypergeometric distribution. See G. K. Miller and Fridell (2007) for more details about the urn problem and the negative hypergeometric

distribution.

The process of proof-of-work mining resembles a classical urn problem in which an urn contains balls (inputs to the SHA-256 function) that result in either a valid hash (success) or a non-valid hash (failure). Each miner is confronted with such an urn and successively tries inputs (which are in fact nonces for a particular potentially valid block) until she has found a valid one (success). Since nonces known to yield a non-valid hash for a particular potentially valid block are not tried twice, this problem resembles the urn problem "without replacement", in which balls representing a failure event are not returned to the urn. Since miners conduct a double-hashing process, the structure of the problem is slightly more complex, while at the end it simplifies to the classical urn problem due to the fact that the SHA-256 function is deterministic. First, a miner tries a nonce for a given potentially valid block (draws a ball) out of the set of 2^{64^2-1} possible elements in the domain of the SHA-256 function. In the context of the urn problem, she first draws a ball from an urn with 2^{64^2-1} balls. The real space from which the miner samples is nevertheless way smaller since it is restricted to the space that contains a potentially valid block. Out of the urn she can get one of 2^{256} results that are contained in the range of the SHA-256 function. We could say that each of these 2^{256} is a different color (or a number). Regardless of the result of the first hash, the miner hashes the result of the first hashing process and gets the final hash that can be either valid or non-valid. Should the resulting second hash be non-valid, the miner will certainly not try again the same nonce, which she now knows yields a non-valid hash. This is the same as not returning the ball representing the non-success event to the urn.

3.3 Data

In this section we describe the data used to empirically study our proposition, and present descriptive statistics that motivate our postulate.

3.3.1 The Bitcoin Blockchain Data

We downloaded all the information about the blockchain available at www.blockchair.com for the blocks 1 to 555,116. These blocks represent the period between January 3, 2009 and December 28, 2018. The raw data contains the following information for each of the blocks: block number, hash of the block, time and date at which the block was mined, the miner (mining pool) that won the block if it is identified, and other metrics about the fees, transactions, and size of the block. An example, for the first and last blocks, is depicted in Table 1.

Block Nr	Hash	Miner	Date	Time
1	000...26f	Unknown	2009-01-03	18:15
2	000...048	Unknown	2009-01-09	02:54
...	...	...	...	...
555,116	000...9b8	AntPool	2018-12-28	15:06

Table 4: Blockchain Data

Due to the nature of the blockchain, we only observe the miner that wins each block. We can observe neither the hash power of the successful miner nor that of the miners that competed to find a valid version of each block but did not succeed in finding it. Since we need the hash power of the miner in order to relate it to the success of winning a block, we construct a proxy for it in the following manner. First, we add a column to the dataset containing the date (natural day), such that we can observe how many blocks were mined each day. This divides our dataset into 3,637 days and not into 3,646 days (the days elapsed between January 3, 2009 and December 28, 2018) due to the fact that on some days (especially at the beginning) no block was mined. Second, we take a sample of the original dataset, containing only the blocks mined after block 278,310, such that we only take into account the blocks mined from January 2, 2014 onward. Our intention in taking this subset is to leave out of our analysis the initial years, during which Bitcoin was not so well spread and no mining pools were formed, and to only start observing the data after ASIC miners were really accessible to the public. Third, we eliminate all the blocks won by an unknown miner. After conducting these steps, we assume that all the miners that have won a block on one day have tried to mine all the blocks that have been mined on that particular day. This allows us to compute the fraction of daily won blocks of each miner. This fraction is our proxy for the fraction of total hash power that each miner uses in each natural day. This proxy contains noise. Since, however, (big) mining pools remain stable and constantly mine blocks without disconnecting their hardware completely, we consider it a sufficiently valid proxy for the purpose of this analysis.

By way of example, we can consider November 12, 2018 (2018-11-12), a day on which 110 blocks were mined by identified miners (blocks mined by unknown miners were set aside). Table 2 summarizes which miners won how many blocks on that day. We observe that the mining pool "AntPool" mined 22 blocks, whereas the mining pool "BTC.COM" mined 21 blocks, the mining pool "F2Pool" 20 blocks, etc. From these frequencies, we can calculate the proportion of blocks won by each miner on each day. For this example, "AntPool" has a fraction of 0.2, which comes from dividing the number of blocks won by it on that day (22) by the total amount of blocks mined that day (110). We take these fractions as a proxy for the daily fraction of total hash power of each miner. While this proxy does not represent the true value of the size of the miner in terms of hash power (the value, in fact, can change on a minute-by-minute or

Date	Pool	Blocks	SIZE	Ranking
2018-11-12	AntPool	22	0.200	1
2018-11-12	BTC.COM	21	0.1909	2
2018-11-12	F2Pool	20	0.1818	3
2018-11-12	SlushPool	14	0.1273	4
2018-11-12	ViaBTC	11	0.1000	5
2018-11-12	BTC.TOP	7	0.0636	6
2018-11-12	BitFury	4	0.0364	7
2018-11-12	Bitcoin.com	3	0.0273	8
2018-11-12	DPOOL	3	0.0273	8
2018-11-12	Bitclub Network	2	0.0182	10
2018-11-12	58COIN	1	0.0091	11
2018-11-12	Eobot	1	0.0091	11
2018-11-12	KanoPool	1	0.0091	11

Table 5: Exemplary data subset for 2018-11-12

second-by-second basis), and it it assumes that hash power is the only factor explaining success in mining (which is exactly the hypothesis that we want to reject), it is the best proxy we have been able to conceive thus far.

From these fractions, which we will hereafter refer to as "size", we can compute the ranking of the miner in terms of size. We order the miners by their daily size, and for each day we assign the ranking with value 1 to the miner with the highest size, the ranking with value 2 to the miner with the second highest size, etc. Should two miners have the same size (i.e., should we observe a tie), both miners get the smallest of the possible rankings. If a tie occurs, the immediately smaller miner after the tied miners gets her true corresponding ranking value (e.g., where both Bitcoin.com and DPOOL share the same size, both receive the ranking value 8 and the Bitclub Network receives the value 10). Having calculated these values for each miner and each day, we incorporate them into the original dataset, which also contains the time required to mine each block, computed as the difference in seconds elapsed between the time of the previous block and the current block.

3.3.2 Descriptive Statistics

Using our dataset, we plot the length of the block in seconds against the ranking of the miner that wins that block. Figure 27 shows the result of this plot. Recall that the biggest miner of the day (i.e., the miner with the estimated highest hash power on a given day) is the one with the ranking equal to 1. Observing this plot it seems that longer blocks are won by relatively bigger pools (i.e., pools with higher ranking places). In other words, it seems that very long blocks are not won by pools that have a lower hash rate per second and that are therefore relatively smaller. If further statistical analysis confirms this

observation, this will imply that there is a "learning" effect within blocks that is proportional to the size of the miner in terms of hash power (i.e., that relatively bigger miners learn faster than relatively smaller miners). One possible explanation for this phenomenon is that the probability of finding a valid block after having found a set of non-valid blocks follows the negative hypergeometric distribution, such that bigger pools' probability of winning increases faster along the length of the block (time in seconds) than smaller pools' probability of winning, due to the fact that relatively bigger miners can try and fail faster (with a higher hash rate per second) than smaller mining pools. Making an analogy with the urn problem, this would mean that bigger pools draw more balls per second without replacement, such that their probability of winning increases with time faster than is the case for relatively smaller miners.

In order to measure this effect, and given the fact that what we consider a "long" block can be arbitrary, we plot—in Figure 28—the probability of the miner with ranking value 1 winning the block given that the block is long. We set different values for what we consider a long block, starting at 575 seconds (the average length of a block) and continuing in steps of 100 seconds until we reach a block time of 4,200 seconds. We observe that the probability of winning of the biggest miner of the day, given that the block is long, tends to increase with the length of the block that we consider "long". This effect becomes visible at a block time of at least 3,275 seconds. In other words, the advantage of the biggest miner starts becoming apparent for blocks of a length equal to or higher than 3,275 seconds. From this length onward, the probability of the miner with ranking 1 winning the block is higher than the probability of the same miner winning any block. These probabilities emerge from the winning frequencies observed in the data and therefore follow no arbitrary decision with regard to what a "long block" is. For illustrative purposes we also plot the probability of the miner with the ranking equal to 1 winning any block and its 99 percent confidence interval. This probability is represented by the straight line at 0.2536.

These plots motivate our work. Our aim is to learn if what we observe in the plots can be explained by the theory postulated in Section 3.2.4 or if, on the contrary, the lack of relatively small miners winning long blocks is just explained by the fact that since small miners by their very definition win less blocks, we have not had enough blocks yet to observe small miners winning long blocks. Should we accept the theory that we postulate in Section 3.2.4, our work could be seen as a tool with which to analyze if the negative hypergeometric distribution that governs the probability of winning blocks can be well approximated by the negative binomial distribution (urn problem with replacement), which is the case when $N, M, N-M \to \infty$, such that $M/N \to p$.

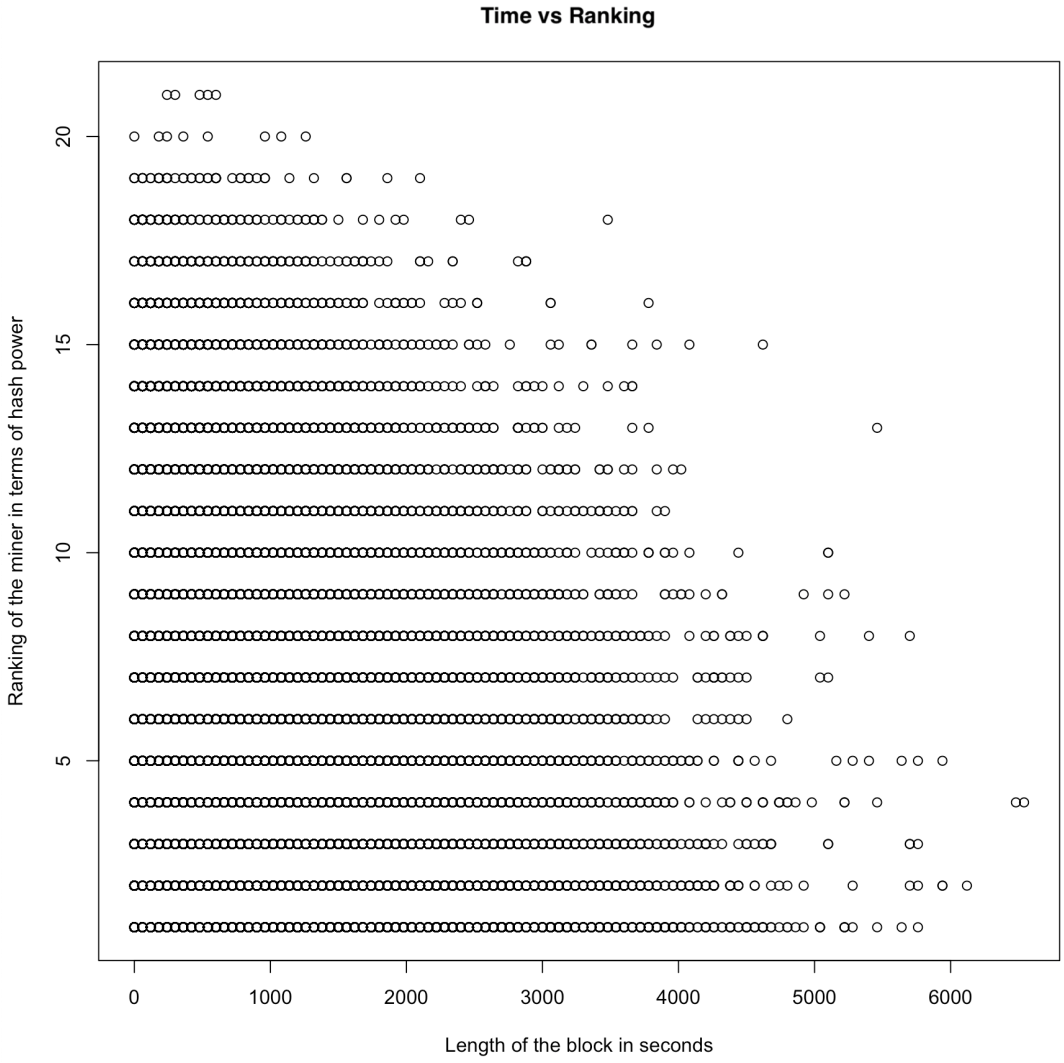

Figure 27: Time required to mine the block vs the ranking of the mining pool in terms of hash power. Period: 2014.01.02 until 2018.12.28.

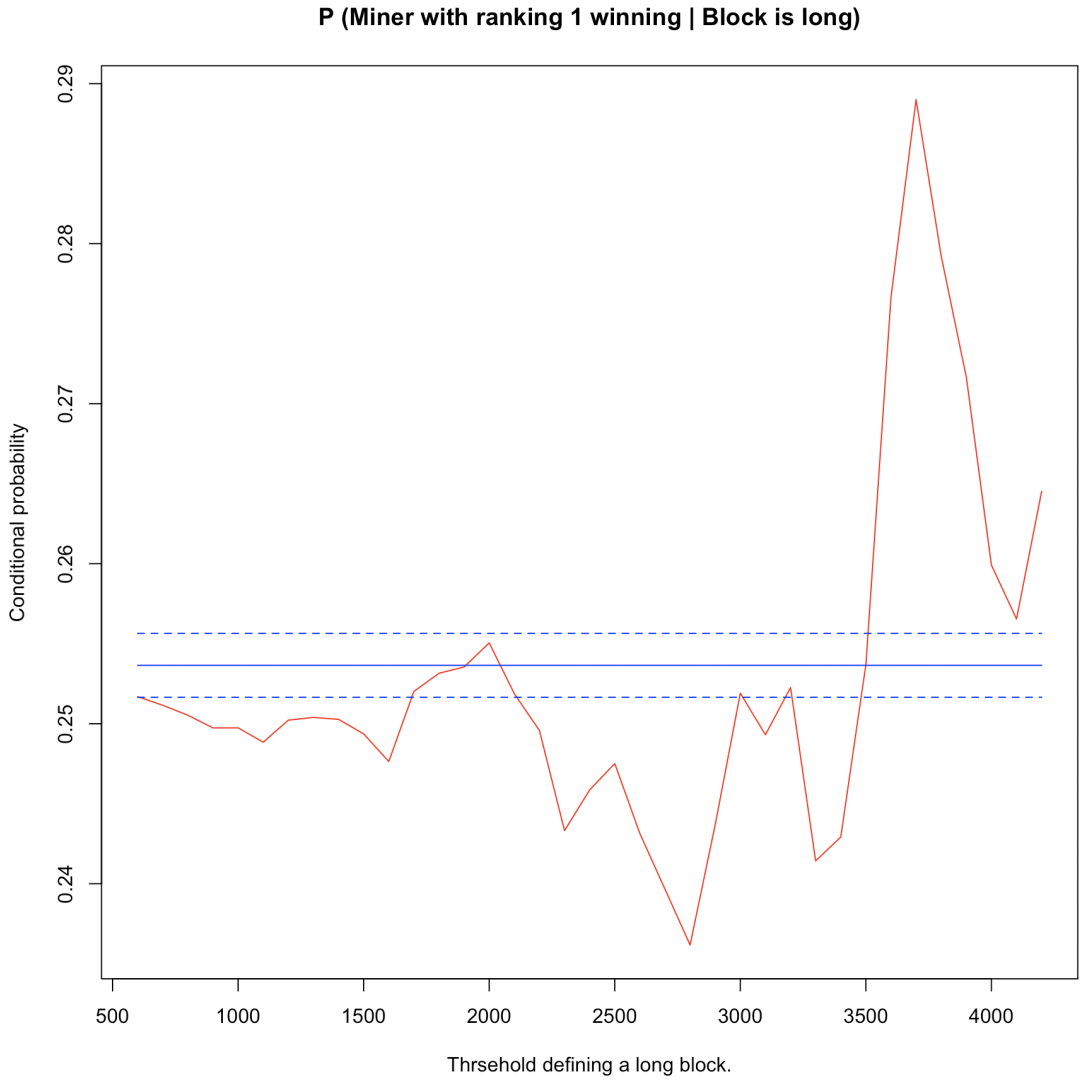

Figure 28: Probability of the biggest miner of the day winning, for different definitions of "long block". Period: 2014.01.02 until 2018.12.28.

3.4 Statistical Model

Races in which only one of the participants can win have been broadly discussed in the literature. Such races have similarities with uncertain future returns of investments. Authors, such as Snyder (1978), Hausch, Ziemba, and Rubinstein (1981), Bolton and Chapman (1986), and Hausch, Lo, and Ziemba (2008), have investigated the properties of horse race markets to determine the impact of the attributes of the horse and of the race, on the probability of winning for each horse in each race. Bolton and Chapman (1986) present a multinomial logit model to analyze the horse race process, recognizing that only a finite number of mutually exclusive outcomes can occur per race—hat is to say, that one, and only one, of the participating horses wins the race. We model the bitcoin mining process as a race with the same structure as the horse race proposed by Bolton and Chapman (1986), such that we can determine the significant impact of attributes of the miners (their size) and attributes of the race (the time required to mine a block) to find out which of those attributes, individually or combined as interaction variables, have an impact on the probability of winning of a particular miner in a particular block.

3.4.1 Stochastic Utility Model of the Mining Process

A bitcoin mining race can be understood as an event in which a decision maker—nature—chooses the winning miner out of a pool of all competing miners. In each block, nature is confronted with a choice set consisting of all the miners mining the block. Each miner i has a vector of S attributes associated with it. In our case, this vector contains the size of the miner, and its ranking, as described in Section 3.3.1, denoted $\boldsymbol{x}_i = [x_{i1}, x_{i2}, ..., x_{iS}]$. The adequate parametrization of this choice model requires that the estimated winning probabilities satisfy the standard axioms of non-negative probabilities. Further, the sum of the probabilities of winninf of all the miners needs to be equal to one. The multinomial logit model described below, which has the same structure as the one presented by Bolton and Chapman (1986), fulfills these requirements. The fact that these requirements need to be fulfilled precludes a simple linear regression model for the estimation since it would violate these probability axioms.

We assume the existence of a utility function U that measures the utility derived for nature by each of the participating miners i winning the block. The utility of a miner in each block can be written as

$$U_i = U(\boldsymbol{x}_i).$$

Since there is always an error in the modelling process, which in our case, given the lack of adequate data, can even be high, the attributes of the vector will not capture all the factors determining the choice.

We therefore assume the overall utility of a miner to have two parts. The first part is the deterministic component $V_i = V(\boldsymbol{x}_i)$. The second part is a random component, $\epsilon_i = \epsilon(\boldsymbol{x}_i)$, which captures the errors in the modelling process. If the stochastic error term is independent of its deterministic component, the utility function U can be written as

$$U_i = \boldsymbol{V}_i + \epsilon_i.$$

Since we have a stochastic error term in this equation, our model is a stochastic utility model. Using this model, let us suppose that miner i^* is observed to win a block. This is the same as observing nature choosing miner i^* out of the set of all miners mining a block. Since we are assuming that nature is rational, nature chooses the miner with the highest utility for this block. Revealed preferences imply that $U_{i^*} \geq U_i$ for $i = 1, 2, ..., I$. Given that the utility function is partly stochastic, the probability of miner i^* winning a block can be written as

$$P_{i^*} = Prob(U_{i^*} \geq U_i, i = 1, 2, ..., I).$$

If we assume that the stochastic error terms are identically and independently distributed (i.i.d.) according to the double exponential distribution $Prob(\epsilon_i \geq \epsilon) = exp[-exp(-\epsilon)]$, then the choice probability assumes the closed-form expression of the multinomial logit model:

$$P_{i^*} = \frac{exp(V_{i^*})}{\sum_{i=1}^{I} exp(V_i)} \qquad \text{for } i^* = 1, 2, ..., I.$$

3.4.2 Estimating the Parameters of the Multinomial Logit Model

The likelihood function associated with a set of blocks can be written for the multinomial logit model as follows:

$$\exp(L) = \prod_{j=1}^{J} P_{jm*},$$

where the subscript j denotes a block ($j = 1, 2, ..., J$), $i*$ is the miner observed to win the block, and L refers to the log-likelihood function.

3.5 Estimation

In this section we describe the data used for the model's estimation, as well as the specification of the model and its results.

3.5.1 Data Used in the Model

The computation of the multinomial logit model requires observations of individuals making a choice about an alternative. Both the individual and the alternatives available to the individual can have attributes that explain the choice. In our context the block is the individual and the winning miner is the alternative. We use the attributes of the alternative (the miner), and only those attributes of the individual (the block) that interact with the attributes of the alternative. This is the exact same principle as the one used by Bolton and Chapman (1986). In order to achieve this, we need to use a dataset containing the miner that won the block as well as all the miners that are assumed to have participated in the mining process of a particular block but did not win that block. This allows us to have different alternatives for each (individual) block. For this computation we use the dataset described in Section 3.3, which contains all blocks won by identified miners for the blocks 278,310 to 555,116.

3.5.2 Specification of the Model

We use the following form of the multinomial logit model in order to model the utility of each miner m:

$$U_i = \theta_1 SIZE_i + \theta_2 SIZE_i * TIME.$$

The variable SIZE_i is the estimated fraction of daily hash power for a miner that we derive as explained in Section 3.3.1. We use SIZE_i as a proxy for the miner's size, measured as a fraction of the total hash power of all miners. The miner's size is assumed to be the main determinant of the mining outcome. Further, we use the variable SIZE_i*TIME, which is the interaction of the block length in seconds with the size of the miner. We include this variable in order to measure the impact on the likelihood of winning of the size across time. In this model, the utility for nature of choosing each alternative depends on the attributes of that alternative (the size of the miner), interacted with the attributes of the individual (the length of the block).

A positive and significant coefficient for the variable SIZE_i*TIME would imply that we can reject that the impact of the miner's size on the likelihood of winning a block does not increase over time. In other words, given the significance of the coefficient SIZE*TIME we can reject the hypothesis that the time elapsed since the moment at which the miner starts mining a block does not increase miners' probability of winning in a manner that is proportional to their size. Hence, we would reject that a miner's probability of winning for a particular block does not increases with the number of previously tried and failed potentially valid blocks (i.e., hashed blocks resulting in a hash above the target). Our postulate would be a possible explanation

for this result. Should the coefficient for the variable SIZE_i*TIME be negative and/or non-significant, this would lead to us not being able to reject that the impact of the miner's size on the likelihood of winning a block does not increase over time and therefore that the observations made in Section 3.3 emerge from the fact that relatively smaller miners have still won too few blocks.

3.5.3 Results of the Estimation

The model was estimated using the blocks described in Section 3.5.1. The associated empirical results are displayed in Table 1. The results of the estimation show both positive and significant coefficients for the variables SIZE_i and SIZE_i*TIME. The coefficients represent the impact of the variable in the log-odds of each alternative being chosen. The estimate for SIZE is positive with a value of 17.223719 and significant with a p-value of 0. This is obvious and was expected, due to the nature of bitcoin mining and also due to the way in which we have built the proxy for the size. From these results we can infer that ceteris paribus, an increase in the size of the miner (i.e., in her hash power) increases her log-odds of winning. The estimate for SIZE*TIME is positive with a value of 0.0001398 and significant with a p-value of 0.0000638. This implies that given the miner's size, we can reject that her log-odds of winning do not increase with the length of the block (i.e., we cannot reject that the log-odds of winning do not increase with time in a manner that is proportional to the size). Since the positive impact of time (block length) on the log-odds of winning interacts with the SIZE, we cannot reject that the log-odds of winning of a bigger miner do not increase with time in a "faster" or "bigger" manner than those of a smaller miner. Such results reveal that time plays a role in miners' probability of winning. A possible explanation of this phenomenon is Proposition 1 and Proposition 2.

Estimation Results			
Attribute	**Estimate**	**Std. Error**	**P($>\|z\|$)**
SIZE	17.223719	0.0277041	0.0000000
SIZE*TIME	0.0001398	0.0000350	0.0000638

3.6 Discussion and Conclusion

The motivation to write this piece emerged while we were studying the "entry–exit" problem that miners face when deciding which cryptocurrency to mine with their hardware. This entry–exit problem is governed by the fixed cost of the mining hardware, the variable electricity cost of using the hardware, the expected return in units of each cryptocurrency that can be mined, and the exchange rate of these cryptocurrencies

to a fiat currency such as the US dollar. Since the expected return of units of the currency is determined by a distribution that depends on the hash power, we started conducting basic descriptive statistics to better understand our problem. These descriptive statistics—summarized in Section 3.3—seemed to contradict the assumptions about the Poisson distribution for the arrival rate of blocks in proof-of-work protocols in general and in bitcoin in particular, an assumption that is well established in the literature and that we describe in Section 3.2. The descriptive statistics suggest that the probability of relatively bigger miners finding longer blocks (longer in terms of the time required to find them) is higher than that of relatively smaller miners. This suggests that there might occur a sort of "learning" when mining a particular block and that relatively bigger miners learn faster than relatively smaller miners. Digging into the literature, we found that recent work by Grunspan and Perez-Marco (2017) and Bowden et al. (2018) has already begun to challenge the assumption of the arrival rate of blocks following the Poisson distribution. However, their respective analyses focus on the security of the proof-of-work protocol and hence on the probability of miners' winning successive blocks. Puzzled by the apparent contradiction between the literature and the statistics emerging from the data, we started revising proof-of-work mining from the basics, postulating that, given a potentially valid block, the number of nonces that needs to be tried by a miner until the first valid block is found is a random variable that follows the negative hypergeometric distribution. This postulate is a possible explanation of the phenomenon observed in the data and is consistent with the technical aspects of bitcoin mining. Further, the resemblance between the urn problem and proof-of-work mining convinces us that this postulate has a theoretical foundation. Should our postulate be correct, it would have serious implications for the way in which scholars and practitioners understand proof-of-work mining. Having postulated this, a question of practical relevance emerged: Does the time that has elapsed since the mining of a block began really play a role in miners' probability of winning?

In order to answer this question, we studied the literature until we found a robust model that would be suitable. Our intention was to measure if the time required to find a block (which represents the number of non-valid nonces tried by a miner for a potentially valid block, given that hash power is measured in hashes per second) had a positive and significant impact on the the likelihood of winning. The model used by Bolton and Chapman (1986) to explain the winning alternative (horse) of a horse race seemed perfectly suited to answering our question since the structure of our problem is the same as that of their problem. The result of this model applied to our problem indicates that we can reject the hypothesis that the size of a miner (her hash power) does not increase her her odds of winning in a way that is proportional to time (i.e., in a way that is proportional to the time spent computing non-valid versions of a potentially valid block). This suggests that the probability of a miner winning a block varies with the number of previously tried and failed attempts for this block and that, therefore, there exists a sort of "learning" within blocks.

Our postulate is a possible explanation for why this is the case.

This result has serious implications for the bitcoin and proof-of-work community. First, it shows that we can still learn more about bitcoin mining. Second, it shows that smaller miners might have an incentive to stop mining a block after trying for a specific time, since after this time their expected reward has decreased so much that it does not compensate their costs. Third, it implies that the way in which platforms are estimating the hash power of mining pools needs to be corrected in order to reflect the impact of time on the success of mining. Fourth, it could be that our postulate explains the concept of mining "luck", since the deviation between the observed and expected performance of miners could be explained by the incorrect approximation of hash power that is used to compute it.

Our results are, though, to be taken with caution. First, since we can only observe the winner of a block, we can only infer which other miners participated in each block. Second, we observe no adequate metrics for hash power and have to derive them from the already inferred and assumed participating miners. Third, the way in which the data is generated implies that the error terms in the model that we use are not i.i.d., which contradicts one of the assumptions of the multinomial logit model.

However, given the theoretical body of our work and the results that emerge from the model—even with a compromised dataset—we conclude that there is enough theoretical and empirical material to sustain that miners' probability of winning does not remain constant over time and that a possible explanation for this phenomenon is that the probability of a miner finding a valid block during a specific attempt follows the negative hypergeometric distribution. This result might have important consequences for the miners of proof-of-work cryptocurrencies in general, and for the miners of bitcoin in particular, as well as for scholars studying cryptocurrencies. We urge the research and practitioner communities to start thinking about mining from this new perspective. Further, we exhort mining pools to report the historical hash power with which they have mined each cryptocurrency, and in a very precise and timely manner. This would allow researchers to continue studying this phenomenon and keep expanding our knowledge of proof-of-work mining.

Part III

Bibliography and Curriculum Vitae

Bibliography

Aggarwal, D., Brennen, G., Lee, T., Santha, M., & Tomamichel, M. (2018). Quantum attacks on bitcoin, and how to protect against them. *Ledger*, *3*(0). Retrieved from https://ledgerjournal.org/ojs/index.php/ledger/article/view/127 doi: 10.5195/ledger.2018.127

Antonopoulos, A. M. (2014). *Mastering bitcoin: Unlocking digital crypto-currencies* (1st ed.). O'Reilly Media, Inc.

Arasa, R. M., & Ottichilo, L. (2015). Determinants of know your customer (kyc) compliance among commercial banks in kenya.. Retrieved from http://hdl.handle.net/11071/2057

Basu, K. (2014). The ponzi economy. *Scientific American*, *310*(6), 70–75. Retrieved from https://www.jstor.org/stable/26039937

Beccuti, J., & Jaag, C. (2017). *The bitcoin mining game: On the optimality of honesty in proof-of-work consensus mechanism* (Working Papers No. 0060). Swiss Economics. Retrieved from https://EconPapers.repec.org/RePEc:chc:wpaper:0060

Beck, R., Avital, M., Rossi, M., & Thatcher, J. B. (2017). Blockchain technology in business and information systems research. *Business & Information Systems Engineering*, *59*(6), 381–384. Retrieved from https://doi.org/10.1007/s12599-017-0505-1 doi: 10.1007/s12599-017-0505-1

Bolton, R. N., & Chapman, R. G. (1986). Searching for positive returns at the track: A multinomial logit model for handicapping horse races. *Management Science*, *32*(8), 1040-1060. Retrieved from https://doi.org/10.1287/mnsc.32.8.1040 doi: 10.1287/mnsc.32.8.1040

Bowden, R., Keeler, H. P., Krzesinski, A. E., & Taylor, P. G. (2018). Block arrivals in the bitcoin blockchain. *CoRR*, *abs/1801.07447*. Retrieved from http://arxiv.org/abs/1801.07447

Britton, M. (2018). *Could blockchain solve the kyc/aml challenge?* Retrieved from https://www.bcsconsulting.com/blog/new-technology-can-enable-human-bank/ (Online; accessed 25 February 2019)

Buckley, R., Arner, D., & Barberis, J. (2016, 01). The emergence of regtech 2.0: From know your customer

to know your data. *Journal of Financial Transformation*, *44*, 79-86. doi: 10.2139/ssrn.3044280

Chiu, J., & Koeppl, T. (2017). *The economics of cryptocurrencies - bitcoin and beyond* (Working Paper No. 1389). Economics Department, Queen's University. Retrieved from https://EconPapers.repec.org/RePEc:qed:wpaper:1389

Civic. (2017). *Secure identity authentication.* Retrieved from https://www.civic.com/developers (Online; accessed 10 August 2018)

Cocco, L., & Marchesi, M. (2016, 10). Modeling and simulation of the economics of mining in the bitcoin market. *PLOS ONE*, *11*(10), 1-31. Retrieved from https://doi.org/10.1371/journal.pone.0164603 doi: 10.1371/journal.pone.0164603

CoinMarketCap. (2019). *Coinmarketcap.* Retrieved from https://coinmarketcap.com (Online; accessed 18 March 2019)

Colladon, A. F., & Remondi, E. (2017). Using social network analysis to prevent money laundering. *Expert Syst. Appl.*, *67*, 49-58.

Cong, L., Li, Y., & Wang, N. (2018). *Tokenomics: Dynamic adoption and valuation* (Working Paper No. 63). Columbia Business School Research Paper. Retrieved from http://dx.doi.org/10.2139/ssrn.3222802 doi: 10.3386/w25592

Cong, L. W., He, Z., & Li, J. (2019). *Decentralized mining in centralized pools* (Working Paper No. 25592). National Bureau of Economic Research. Retrieved from http://www.nber.org/papers/w25592 doi: 10.3386/w25592

Copeland, T. E., & Weston, J. (1979). *Financial theory and corporate policy.*

Courtois, N. T., Grajek, M., & Naik, R. (2014a). Optimizing sha256 in bitcoin mining. In Z. Kotulski, B. Kotulski, & K. Mazur (Eds.), *Cryptography and security systems* (pp. 131–144). Berlin, Heidelberg: Springer Berlin Heidelberg.

Courtois, N. T., Grajek, M., & Naik, R. (2014b). The unreasonable fundamental incertitudes behind bitcoin mining. *CoRR*, *abs/1310.7935*. Retrieved from http://arxiv.org/abs/1310.7935

Decker, C., & Wattenhofer, R. (2013). Information propagation in the bitcoin network. *IEEE P2P 2013 Proceedings*, 1-10.

Dimitri, N. (2017). Bitcoin mining as a contest. *Ledger*, *2*(0), 31–37. Retrieved from https://ledgerjournal.org/ojs/index.php/ledger/article/view/96 doi: 10.5195/ledger.2017.96

Easley, D., O'Hara, M., & Basu, S. (2019, 5). From mining to markets: The evolution of bitcoin transaction fees. *Journal of Financial Economics (JFE), Forthcoming*. doi: http://dx.doi.org/10.2139/ssrn.3055380

Egelund-Müller, B., Elsman, M., Henglein, F., & Ross, O. (2017). Automated execution of financial contracts on blockchains. *Business & Information Systems Engineering*, *59*(6), 457–467. Retrieved

from https://doi.org/10.1007/s12599-017-0507-z doi: 10.1007/s12599-017-0507-z

European Central Bank. (2012). *Virtual currency schemes.* Retrieved from https://www.ecb.europa.eu/pub/pdf/other/virtualcurrencyschemes201210en.pdf (Online; accessed 31 October 2017)

European Commission. (2012). *Reform of eu data protection rules- european commission.* Retrieved from http://ec.europa.eu/justice/data-protection/reform/index_en.htm (Online; accessed 31 October 2017)

European Security and Markets Authority. (2016). *The distributed ledger technology applied to securities markets.* Retrieved from https://www.esma.europa.eu/sites/default/files/library/2016-773_dp_dlt.pdf (Online; accessed 31 October 2017)

Eyal, I., & Sirer, E. G. (2013). Majority is not enough: Bitcoin mining is vulnerable. *CoRR*, *abs/1311.0243*. Retrieved from http://arxiv.org/abs/1311.0243

Foley, S., Karlsen, J. R., & Putniņš, T. J. (2019). Sex, Drugs, and Bitcoin: How Much Illegal Activity Is Financed through Cryptocurrencies? *The Review of Financial Studies*, *32*(5), 1798-1853. Retrieved from https://doi.org/10.1093/rfs/hhz015 doi: 10.1093/rfs/hhz015

Freifeld, K. (2012). *Ing to pay $619 million over cuba, iran sanctions.* Retrieved from http://www.reuters.com/article/us-ing-sanctions-idUSBRE85B12I20120612 (Online; accessed 31 October 2017)

Friedman, M. (1999). *How economist milton friedman predicted bitcoin.* Retrieved from https://www.coindesk.com/economist-milton-friedman-predicted-bitcoin (Online; accessed 29 April 2019)

Glaser, F. (2017). *Pervasive decentralisation of digital infrastructures: A framework for blockchain enabled system and use case analysis.* 50th Hawaii International Conference on System Sciences (HICSS 2017); Waikoloa Village, Hawaii, USA, 3 - 7 January 2017.

Göbel, J., Keeler, H. P., Krzesinski, A. E., & Taylor, P. G. (2015). Bitcoin blockchain dynamics: the selfish-mine strategy in the presence of propagation delay. *CoRR*, *abs/1505.05343*. Retrieved from http://arxiv.org/abs/1505.05343

GrandViewResearch. (2018). *Blockchain technology market size, share and trends analysis report by type (public, private, hybrid), by application (financial services, consumer products, technology, telecom), and segment forecasts, 2018 - 2024.*

Grunspan, C., & Pérez-Marco, R. (2017). Double spend races. *CoRR*, *abs/1702.02867*. Retrieved from http://arxiv.org/abs/1702.02867

Hanke, T. (2016). Asicboost - A speedup for bitcoin mining. *CoRR*, *abs/1604.00575*. Retrieved from http://arxiv.org/abs/1604.00575

Hausch, D., Lo, V., & Ziemba, W. T. (2008). *Efficiency of racetrack betting markets (2008 edition).* World Scientific Publishing Company. Retrieved from https://books.google.ch/books

?id=8ATGCgAAQBAJ

Hausch, D., Ziemba, W., & Rubinstein, M. (1981, 12). Efficiency of the market for racetrack betting. *Management Science*, *27*, 1435-1452. doi: 10.1287/mnsc.27.12.1435

Hayes, A. S. (2019). Bitcoin price and its marginal cost of production: support for a fundamental value. *Applied Economics Letters*, *26*(7), 554-560. Retrieved from https://doi.org/10.1080/13504851.2018.1488040 doi: 10.1080/13504851.2018.1488040

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Q.*, *28*(1), 75–105. Retrieved from http://dl.acm.org/citation.cfm?id=2017212.2017217

Hong Kong Monetary Authority. (2017). *Whitepaper 2.0. on distributed ledger technology.* Retrieved from https://www.hkma.gov.hk/media/eng/doc/key-functions/finanical-infrastructure/infrastructure/20171025e1a1.pdf (Online; accessed 25 February 2019)

Houy, N. (2016). The bitcoin mining game. *Ledger*, *1*(0), 53–68. Retrieved from http://ledger.pitt.edu/ojs/index.php/ledger/article/view/13 doi: 10.5195/ledger.2016.13

IEEE Computer Society. (2000). *Ieee standard specifications for public-key cryptography.* Retrieved from https://perso.telecom-paristech.fr/guilley/recherche/cryptoprocesseurs/ieee/00891000.pdf

Johnson, N. L., & Kotz, S. (1969). *Distributions in statistics: Discrete distributions.* The Houghton Mifflin Series in Statistics. Boston: Houghton Mifflin Company. XVI, 328 p. $ 12.50 (1969).

Karlstrøm, H. (2014). Do libertarians dream of electric coins? the material embeddedness of bitcoin. *Distinktion: Journal of Social Theory*, *15*(1), 23-36. Retrieved from https://doi.org/10.1080/1600910X.2013.870083 doi: 10.1080/1600910X.2013.870083

Krugman, P. (2013). *Bitcoin is evil.* Retrieved from https://krugman.blogs.nytimes.com/2013/12/28/bitcoin-is-evil/ (Online; accessed 29 April 2019)

Lee, K., James, J. I., Ejeta, T. G., & Kim, H. J. (2016). Electronic voting service using block-chain. *JDFSL*, *11*(2), 123–136. Retrieved from http://ojs.jdfsl.org/index.php/jdfsl/article/view/414

Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., & Rosenschein, J. S. (2015). Bitcoin mining pools: A cooperative game theoretic analysis. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems* (pp. 919–927). Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. Retrieved from http://dl.acm.org/citation.cfm?id=2772879.2773270

Lindman, M. T. V., Juho; Rossi. (2017). Opportunities and risks of blockchain technologies in payments – a research agenda [A4 Artikkeli konferenssijulkaisussa]. In (p. 10; 1533-1542). Retrieved from http://urn.fi/URN:NBN:fi:aalto-201711217679 doi: 10.24251/HICSS.2017.185

Liu, Z., Luong, N. C., Wang, W., Niyato, D., Wang, P., Liang, Y.-C., & Kim, D. I. (2019). A survey on

applications of game theory in blockchain. *CoRR*, *abs/1902.10865*.

Luu, L., Chu, D.-H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. *IACR Cryptology ePrint Archive*, *2016*, 633.

Memminger M, Baxter M, Lin E. (2016). *Banking regtechs to the rescue?* Retrieved from `http://www.bain.com/Images/BAIN_BRIEF_Banking_Regtechs_to_the_Rescue.pdf` (Online; accessed 31 October 2017)

Miller, A. K., & LaViola, J. J. (2014). Byzantine consensus from moderately-hard puzzles : A model for bitcoin..

Miller, G. K., & Fridell, S. L. (2007). A forgotten discrete distribution? reviving the negative hypergeometric model. *The American Statistician*, *61*(4), 347–350. Retrieved from `http://www.jstor.org/stable/27643937`

Nakamoto, S. (2008). *Bitcoin: A peer-to-peer electronic cash system.* Retrieved from `https://bitcoin.org/bitcoin.pdf` (Online; accessed 18 March 2019)

Palmer, D. (2014). *Economist nouriel roubini slams bitcoin, calls it a 'ponzi game'.* Retrieved from `https://www.coindesk.com/economist-nouriel-roubini-slams-bitcoin-calls-ponzi-game` (Online; accessed 29 April 2019)

Parra-Moyano, J., & Ross, O. (2017). Kyc optimization using distributed ledger technology. *Business & Information Systems Engineering*, *59*(6), 411–423. Retrieved from `https://doi.org/10.1007/s12599-017-0504-2` doi: 10.1007/s12599-017-0504-2

Parra-Moyano, J., & Schmedders, K. (2018). *The liberalization of data: A welfare-enhancing information system.* Retrieved from `https://ssrn.com/abstract=3302752`

Parra-Moyano, J., Thoroddsen, T., & Ross, O. (2019). Optimized and dynamic kyc system based on blockchain technology. *International Journal of Blockchains and Cryptocurrencies*, *1*.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007, 01). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*, 45-77.

Peters, G. W., Panayi, E., & Chapelle, A. (2015). Trends in crypto-currencies and blockchain technologies: A monetary theory and regulation perspective. *CoRR*, *abs/1508.04364*. Retrieved from `http://arxiv.org/abs/1508.04364`

R. Harvey, C. (2014). Cryptofinance. *SSRN Electronic Journal*. doi: 10.2139/ssrn.2438299

Rosenfeld, M. (2011). Analysis of bitcoin pooled mining reward systems. *CoRR*, *abs/1112.4980*. Retrieved from `http://arxiv.org/abs/1112.4980`

Rosenfeld, M. (2014). Analysis of hashrate-based double spending. *CoRR*, *abs/1402.2009*. Retrieved from `http://arxiv.org/abs/1402.2009`

Ruce, P. J. (2011). Anti-money laundering: The challenges of know your customer legislation for private bankers and the hidden benefits for relationship management ('the bright side of knowing your customer'). *The Banking Law Journal*, *128*. Retrieved from http://arxiv.org/abs/1508.04364

Rutter, K. (2018). *If at first you don't succeed, try a decentralized kyc platform.* Retrieved from https://www.r3.com/wp-content/uploads/2018/10/first_succeed_decentralized_R3.pdf (Online; accessed 25 February 2019)

Sapirshtein, A., Sompolinsky, Y., & Zohar, A. (2015). Optimal selfish mining strategies in bitcoin. *CoRR*, *abs/1507.06183*. Retrieved from http://arxiv.org/abs/1507.06183

Shocard. (2017). *How it works.* Retrieved from https://shocard.com/how-it-works (Online; accessed 10 August 2018)

Siegenthaler, M., & Birman, K. (2009a, April). Privacy enforcement for distributed healthcare queries. In *2009 3rd international conference on pervasive computing technologies for healthcare* (p. 1-6). doi: 10.4108/ICST.PERVASIVEHEALTH2009.6016

Siegenthaler, M., & Birman, K. (2009b, July). Sharing private information across distributed databases. In *2009 eighth ieee international symposium on network computing and applications* (p. 82-89). doi: 10.1109/NCA.2009.33

Smet, D., & Mention, A. (2011). Improving auditor effectiveness in assessing kyc/aml practices: Case study in a luxembourgish context. *Managerial Auditing Journal*, *26*(2), 182-203. Retrieved from https://doi.org/10.1108/02686901111095038 doi: 10.1108/02686901111095038

Snyder, W. W. (1978). Horse racing: Testing an efficient markets model. *The Journal of Finance*, *33*(4), 1109-1118. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1978.tb02051.x doi: 10.1111/j.1540-6261.1978.tb02051.x

Solat, S., & Potop-Butucaru, M. (2016). Zeroblock: Preventing selfish mining in bitcoin. *CoRR*, *abs/1605.02435*. Retrieved from http://arxiv.org/abs/1605.02435

Soni, A., & Duggal, R. (2014, 07). Reducing risk in kyc (know your customer) for large indian banks using big data analytics. *International Journal of Computer Applications*, *97*, 49-53. doi: 10.5120/17039-7347

Stross, C. (2013). *Why i want bitcoin to die in a fire.* Retrieved from http://www.antipope.org/charlie/blog-static/2013/12/why-i-want-bitcoin-to-die-in-a.html (Online; accessed 29 April 2019)

Szabo, N. (1997). Smart contracts: formalizing and securing relation- ships on public networks. *Expert Syst Appl*, *2*. Retrieved from https://journals.uic.edu/ojs/index.php/fm/article/view/548

Thompson Reuters. (2016). *Know your customer (kyc) independent survey.* Retrieved from https://www.refinitiv.com/en/resources/infographics/2016-know-your-customer-kyc

-independent-survey/ (Online; accessed 31 October 2017)

Tth2549. (2017). *Kyc-optimized-and-dynamic-system-using-blockchain-technology.* Retrieved from https://github.com/tth2549/KYC-Optimized-and-Dynamic-System-using-Blockchain-Technology (Online; accessed 12 June 2018)

UBS. (2016). *Utility settlement coin concept on blockchain gathers pace.* Retrieved from https://bit.ly/2BWuM5U (Online; accessed 10 August 2018)

USA. (1986). *The money laundering control act of 1986 (public law 99-570).*

USA. (1988). *The anti-drug abuse act of 1988, (public law law 100690).*

USA. (1992). *Annunzio-wylie anti-money laundering act of 1992.*

USA. (1994). *The money laundering suppression act of 1994.*

USA. (1998). *The money laundering and financial crimes act.*

USA. (2001). *Usa patriot act of 2001 (public law 107-156).*

USA. (2004). *Intelligence reform and terrorism prevention act of 2004.*

Viswanatha A. and Wolf B. (2012). *Hsbc to pay $1.9 billion u.s. fine in money-laundering case.* Retrieved from http://www.reuters.com/article/us-hsbc-probe-idUSBRE8BA05M20121211 (Online; accessed 31 October 2017)

Voglsteller, F. and Buterin, V. (2015). *Erc-20 token standard.* Retrieved from https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md (Online; accessed 10 August 2018)

Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., ... Kim, D. I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, *7*, 22328-22370. doi: 10.1109/ACCESS.2019.2896108

Wood G. (2016). *Ethereum: a secure decentralised generalised transaction ledger.* Retrieved from http://gavwood.com/paper.pdf (Online; accessed 31 October 2017)

World Economic Forum. (2016). *The future of financial infrastructure.* Retrieved from http://www3.weforum.org/docs/WEF_The_future_of_financial_infrastructure.pdf. (Online; accessed 31 October 2017)

Curriculum Vitæ

José Parra Moyano

Personal Information

Date and Place of Birth:	February 5th, 1991 in Madrid, Spain
Place of Residence	Zurich, Switzerland

Education

08/16 - 09/19	**PhD Studies in Management and Economics** University of Zurich, Chair of Quantitative Business Administration Advisors: Prof. Dr. Karl Schmedders, and Prof. Dr. Claudio J. Tesone Award: 1st Prize, Nordic Blockchain Summit 2016 Mention: Summa Cum Laude
09/13 - 07/16	**Master of Arts in Economics** University of Zurich Mention: Magna Cum Laude
09/09 - 10/12	**Bachelor of Arts in Economics** University of Zurich Mention: Cum Laude
09/08 - 05/09	**Matura** Swiss School of Madrid Undergraduate Education